

UNIVERSIDAD POLITÉCNICA ESTATAL DEL CARCHI



FACULTAD DE INDUSTRIAS AGROPECUARIAS Y CIENCIAS AMBIENTALES CARRERA DE INGENIERÍA EN INFORMÁTICA

Tema: “Propuesta de un Sistema de Gestión de Inventarios para la Farmacia del Hospital del Dia Nova Salud”

Trabajo de titulación previa la obtención del
Título de Ingeniero en Informática

AUTOR: Darwin Fernando Tipan Viveros

TUTOR: Msc. Samuel Benjamín Lascano Rivera

Tulcán, 2021

CERTIFICADO JURADO EXAMINADOR

Certificamos que el estudiante Darwin Fernando Tipan Viveros con el número de cédula 1725098899 ha elaborado el trabajo de titulación: “Propuesta de un Sistema de Gestión de Inventarios para la Farmacia del Hospital del Dia Nova Salud”

Este trabajo se sujeta a las normas y metodología dispuesta en el Reglamento de Titulación, Sustentación e Incorporación de la UPEC, por lo tanto, autorizamos la presentación de la sustentación para la calificación respectiva.

f.....

Msc. Samuel Benjamín Lascano Rivera

TUTOR

f.....

Msc. Jeffery Naranjo

LECTOR

Tulcán, Marzo 2022

AUTORÍA DE TRABAJO

El presente trabajo de titulación constituye un requisito previo para la obtención del título de **Ingeniero** en la Carrera de ingeniería en informática de la Facultad de Industrias Agropecuarias y Ciencias Ambientales

Yo, Darwin Fernando Tipan Viveros con cédula de identidad número 1725098899 declaró: que la investigación es absolutamente original, auténtica, personal y los resultados y conclusiones a los que he llegado son de mi absoluta responsabilidad.

f.....

Darwin Fernando Tipan Viveros

AUTOR

Tulcán, Marzo 2022

ACTA DE CESIÓN DE DERECHOS DEL TRABAJO DE TITULACIÓN

Yo, Darwin Fernando Tipan Viveros declaro ser autor/a de los criterios emitidos en el trabajo de investigación: “Propuesta de un Sistema de Gestión de Inventarios para la Farmacia del Hospital del Dia Nova Salud” y eximo expresamente a la Universidad Politécnica Estatal del Carchi y a sus representantes legales de posibles reclamos o acciones legales.

f.....

Darwin Fernando Tipan Viveros

AUTOR

Tulcán, Marzo 2022

AGRADECIMIENTO

Agradecer al establecimiento de Salud “Hospital del Dia Nova Salud” por abrirme sus puertas para realizar mi trabajo de titulación.

Agradecer a la UNIVERSIDAD POLITÉCNICA ESTATAL DE CARCHI por darme una gran oportunidad de ingresar a estudiar a su institución y formarme como un gran profesional, al igual de brindarme valores éticos y morales que los aplicare en el transcurso de mi vida profesional y cotidiana.

Agradecer al esfuerzo que han hecho mis padres, Gladys Fabiola Viveros Montenegro, Luis Alberto Tipan Valarezo, al apoyarme a culminar mis estudios, a pesar de todas las adversidades que me han ayudado a superar, conjunto con el cariño de mis hermanos, Jefferson Bladimir Tipan Viveros, Elizabeth Margarita Diaz Viveros.

Agradecer a mi abuelita Laurentina Montenegro y a mi tío Alfredo Montenegro, que están sentados a lado de dios, por acompañarme desde el cielo y dándome su apoyo celestial en todo momento.

Agradecer a la divinidad de la virgencita de la paz, por ayudarme a cumplir mi sueño de ser un profesional para llenar de orgullo a mis padres y a todos mis seres queridos.

Agradecerles a mis amigos más cercanos, Bryan Drouet, Kevin Paucar, Jefferson Portilla, Jhonson García, Nina Maigua, Cytia Mashinkash, Santiago Chipantiza, son las personas que me han apoyado en esta etapa de mi vida, donde fueron parte de ella en el transcurso de los 5 años de carrera y lo seguirán haciendo hasta la eternidad.

Agradecer a mi mentor y lector, Samuel Lascano, Jeffery Naranjo, por ser parte del trascurso al realizar mi trabajo de titulación, donde fueron más que un profesor, fueron amigos y grandes personas al brindarme su conocimiento y su apoyo en todo momento.

Por último, agradecerme a mi persona por ser fuerte, resistente y no dejarme vencer por las etapas que viví lejos de casa en el transcurso de mi carrera, de luchar por la promesa que les hice a mis padres desde que empezó esta travesía y hoy la estoy cumpliendo.

DEDICATORIA

Este logro lo dedico a las dos personas que me dieron la oportunidad de vivir en este mundo tan maravilloso y teniendo la dicha y el orgullo de ser su hijo.

También agradecer a mi familia por el apoyo incondicional que me dieron para culminar de la mejor manera esta etapa de mi vida, donde empezaré a ver la vida de una forma distinta a la que conocía.

Dedico este logro a las dos personas más importantes en mi vida:

A mi madre:

Por enseñarme que todo lo que se propone en esta vida se puede lograr, por ser la persona que me ayudó a comprender de una forma distinta la vida, por ser una amiga una hermana y por ser una gran mamá en toda mi etapa de crecimiento. “TE AMO mamita, Gladys Fabiola Viveros Montenegro”.

A mi padre:

Por inculcarme valores desde pequeño, hasta ser un adulto con carácter y fortalezas, donde me enseñó que no hay que rendirse a pesar de las circunstancias que pasemos en la vida, “TE AMO papito, Luis Alberto Tipan Valarezo”.

ÍNDICE

I. PROBLEMA	23
1.1. PLANTEAMIENTO DEL PROBLEMA	23
1.2. FORMULACIÓN DEL PROBLEMA	24
1.3. JUSTIFICACIÓN	25
1.4. OBJETIVOS Y PREGUNTAS DE INVESTIGACIÓN	25
1.4.1. Objetivo General	25
1.4.2. Objetivos Específicos	25
1.4.3. Preguntas de Investigación	26
II. FUNDAMENTACIÓN TEÓRICA	27
2.1. ANTECEDENTES INVESTIGATIVOS	27
2.2. MARCO TEÓRICO	29
2.2.1. Control de inventario	29
2.2.2. Tipos de inventario	30
2.2.3. Clasificación concepción logística	31
2.2.3.1. Inventarios de lote o de ciclos	31
2.2.3.2. Inventarios de seguridad	31
2.2.4. Clasificación impulsada por la demanda	32
2.2.5. Tipos de modelos de inventarios	33
2.2.6. La gestión del inventario	34
2.2.7. Definición de stock	34
2.2.7.1. Stock óptimo	35
2.2.7.2. Stock de seguridad	35
2.2.7.3 Stock circulante	35
2.2.7.4. Stock ciego	35
2.2.7.5. Stock vivo	35
2.2.7.6. Stock ideal	35

2.2.8. Métodos de valoración de inventario	35
2.2.8.1. Método FIFO o PEPS	36
2.2.8.2. Método LIFO O UEPS	36
2.2.8.3. Método de costo promedio aritmético	36
2.2.8.4. Método del promedio armónico o ponderado	36
2.2.8.5. Método del costo promedio móvil o del saldo	36
2.2.8.6. Método del costo básico	36
2.2.8.7. Método del precio de venta al detal	36
2.2.8.8. Costo de mercado o el más bajo	36
2.2.9 Software de control de inventario	37
2.2.9.1. Tipos de software de manejo de inventario	37
2.2.9.2. Software de inventario perpetuo	37
2.2.9.3. Software de inventario periódico	38
2.2.9.4. Software de inventario de código de barras	38
2.2.9.5. Software de inventario de Identificación por Radiofrecuencia (RFID)	39
2.2.10. La Ingeniería del software en el control de inventario	40
2.2.11. Metodologías de desarrollo de software	40
2.2.11.1. Metodología Adaptive Software Development	41
2.2.11.2. Metodología Scrum	42
2.2.11.3. Metodología XP	42
2.2.12. Tipos de arquitectura de desarrollo de software	46
2.2.12.1. Arquitectura cliente servidor	47
2.2.12.2. Arquitectura por capas	48
2.2.12.3. Arquitectura Modelo-vista-controlador	49
2.2.13. Lenguajes de programación	54
2.2.13.1. Lenguaje Java	54
2.2.13.2. Lenguaje Python	54

2.2.13.3. Lenguaje PhP	55
2.2.14. Framework de desarrollo web relacionados con PHP	56
2.2.14.1. Framework Laravel	56
2.2.14.2. Framework Slim	56
2.2.14.3. Framework Codeigniter V4	57
2.2.15. Gestores de bases de datos	58
2.2.15.1. Gestor de base de datos PostgreSQL	58
2.2.15.2. Gestor de base de datos SQLite	58
2.2.15.3. Gestor de base de datos MySQL	59
2.2.16. Análisis de las herramientas y diseño de un sistema	61
III. METODOLOGÍA	62
3.1. ENFOQUE METODOLÓGICO	62
3.1.1. Enfoque cualitativo	62
3.1.2. Tipo de investigación	62
3.2. IDEA A DEFENDER	63
3.3. Operacionalización de variables	64
3.4. MÉTODOS UTILIZADOS	66
3.4.1. Investigación cuasi experimental	66
3.4.2. Técnicas	66
IV. RESULTADOS Y DISCUSIÓN	68
4.1. RESULTADOS	68
4.2. DISCUSIÓN	90
V. CONCLUSIONES Y RECOMENDACIONES	92
5.1. CONCLUSIONES	92
5.2. RECOMENDACIONES	93
IV. REFERENCIAS BIBLIOGRÁFICAS	94
V. ANEXOS	102

ÍNDICE DE FIGURAS

Figura 1. Modelo de control de inventario	29
Figura 2. Beneficio de las metodologías ágiles	41
Figura 3. Fases de desarrollo metodología XP	45
Figura 4. Función de la arquitectura	47
Figura 5. Función de la arquitectura	48
Figura 6. Función del modelo	49
Figura 7. Función de la vista	50
Figura 8. Función del controlador	50
Figura 9. Arquitectura de Modelo Vista Controlador	52
Figura 10. Logotipo de framework de laravel	56
Figura 11. Proceso de adquisición de productos	70
Figura 12. Proceso de devolución y recibimiento de los M/A	72
Figura 13. Diseño UML	188
Figura 14. Diseño de base de datos	189
Figura 15. Maquetación de inicio de sesión, administrador	190
Figura 16. Maquetación de panel de control	190
Figura 17. Maquetación de ventas	191
Figura 18. Maquetación de compras	191
Figura 19. Maquetación de catálogos	192
Figura 20. Maquetación de herramientas	192
Figura 21. Maquetación de finanzas	193
Figura 22. Maquetación de inventario	193
Figura 23. Maquetación de reportes	194
Figura 24. Maquetación de administración	194
Figura 25. Maquetación de inicio de sesión	195

Figura 26. Maquetación de ventas	195
Figura 27. Maquetación de compras	196
Figura 28. Maquetación de inventario	196
Figura 29. Maquetación de inicio de sesión	197
Figura 30. Maquetación de ventas	197
Figura 31. Maquetación de inventario, buscar producto, devolución	198
Figura 32. Diagrama de caso de uso administrador/inicio	198
Figura 33. Diagrama de caso de uso administrador/notificaciones	198
Figura 34. Diagrama de caso de uso administrador/vender	199
Figura 35. Diagrama de caso de uso administrador/ventas	199
Figura 36. Diagrama de caso de uso administrador/compras	199
Figura 37. Diagrama de caso de uso administrador/catálogos	200
Figura 38. Diagrama de caso de uso administrador/herramientas	200
Figura 39. Diagrama de caso de uso administrador/finanzas	200
Figura 40. Diagrama de caso de uso administrador/inventario	201
Figura 41. Diagrama de caso de uso administrador/reportes	201
Figura 42. Diagrama de caso de uso administrador/administrador	202
Figura 43. Diagrama de caso de uso vendedor/inicio	202
Figura 44. Diagrama de caso de uso vendedor/vender	202
Figura 45. Diagrama de caso de uso vendedor/ventas	203
Figura 46. Diagrama de caso de uso vendedor/inventario	203
Figura 47. Diagrama de caso de uso vendedor/buscar producto	203
Figura 48. Diagrama de caso de uso vendedor/devoluciones	203
Figura 49. Diagrama de caso de uso bodeguero /inicio	204
Figura 50. Diagrama de caso de uso bodeguero/vender	204
Figura 51. Diagrama de caso de uso bodeguero/ compras	204
Figura 52. Diagrama de caso de uso bodeguero/inventario	204

Figura 53. Codificación de modelos	205
Figura 54. Codificación de los modelos	205
Figura 55. Codificación de los modelos	206
Figura 56. Codificación de modelos	206
Figura 57. Codificación de modelos	207
Figura 58. Codificación de vistas	207
Figura 59. Codificación de vistas	208
Figura 60. Codificación de vistas	208
Figura 61. Codificación de vistas	209
Figura 62. Codificación de vistas	209
Figura 63. Codificación de vistas	210
Figura 64. Codificación de controladores	210
Figura 65. Codificación de controladores	211
Figura 66. Codificación de controladores	211
Figura 67. Codificación de controladores	212
Figura 68. Codificación de controladores	212
Figura 69. Codificación de controladores	213
Figura 70. Inicio de sesión	231
Figura 71. Panel principal del sistema	231
Figura 72. Módulo de categorías	232
Figura 73. Sección para agregar nuevo producto en el sistema	232
Figura 74. Sección para visualizar los productos registrados	233
Figura 75. Módulo del inventario	233
Figura 76. Sección de realización de ventas	234
Figura 77. Sección de caja	234
Figura 78. Sección de reabastecimiento del inventario	235
Figura 79. Sección de compras	235

Figura 80. Lista de usuarios	236
Figura 81. Inicio de sesión	237
Figura 82. Panel de control	237
Figura 83. Ventana de agregación de un nuevo producto	238
Figura 84. Venta del inventario	238
Figura 85. Ventana de ventas	239
Figura 86. Ventana de reabastecimiento del inventario	239
Figura 87. Inicio de sesión	240
Figura 88. Ventana de ventas	240
Figura 89. Ventana de inventario	241
Figura 90. Ventana de búsqueda entre sucursales	241
Figura 91. Ventana de devoluciones	242

ÍNDICE DE TABLAS

Tabla 1. Clasificación de inventarios ABC	32
Tabla 2. Comparación de metodologías de desarrollo	43
Tabla 3. Comparación de arquitecturas de desarrollo de software	51
Tabla 4. Comparación de los lenguajes de programación	55
Tabla 5. Comparación de framework de desarrollo web	57
Tabla 6. Comparación de gestores de bases de datos	60
Tabla 7. Operacionalización de la variable independiente "Sistema"	64
Tabla 8. Operacionalización de la variable dependiente "Gestión del Inventario"	65
Tabla 9. Primera pregunta: Administradora	69
Tabla 10. Segunda pregunta: Administradora	71
Tabla 11. Tercera pregunta: Administradora	73
Tabla 12. Cuarta pregunta: Administradora	74
Tabla 13. Quinta pregunta: Administradora	75
Tabla 14. Sexta pregunta: Administradora	76
Tabla 15. Séptima pregunta: Administradora	77
Tabla 16. Octava pregunta: Administradora	78
Tabla 17. Novena pregunta: Administradora	79
Tabla 18. Primera pregunta: Bodeguero	80
Tabla 19. Segunda pregunta: Bodeguero	81
Tabla 20. Tercera pregunta: Bodeguero	82
Tabla 21. Cuarta pregunta: Bodeguero	82
Tabla 22. Quinta pregunta: Bodeguero	83
Tabla 23. Sexta pregunta: Bodeguero	83
Tabla 24. Séptima pregunta: Bodeguero	84
Tabla 25. Matriz de Sp 1.1	85
Tabla 26. Matriz de resultado Sp 1.2	86

Tabla 27. Matriz de respuesta por porcentajes	86
Tabla 28. Matriz de resultado final de las áreas de proceso	86
Tabla 29. Hoja técnica	87
Tabla 30. Ficha de observación del manejo de inventario	88
Tabla 31. Historial de revisiones	149
Tabla 32. Recursos humanos y profesionales	150
Tabla 33. Roles y responsabilidades	150
Tabla 34. Plan de proyecto	151
Tabla 35. Fases y líneas base	151
Tabla 36. Objetivos y riesgos	152
Tabla 37. Diagrama de Gantt	152
Tabla 38. Calendario de proyecto	153
Tabla 39. Control de desviación a la planificación	153
Tabla 40. Matriz de riesgos	154
Tabla 41. Matriz de calor	154
Tabla 42. Recursos de software	155
Tabla 43. Recursos de hardware	156
Tabla 44. Costos de equipos	156
Tabla 45. Recursos de oficina	156
Tabla 46. Roles del proyecto	157
Tabla 47. Estimación de tiempo	158
Tabla 48. Historia de usuario 1	160
Tabla 49. Historia de usuario 2	160
Tabla 50. Historia de usuario 3	161
Tabla 51. Historia de usuario 4	162
Tabla 52. Historia de usuario 5	162
Tabla 53. Historia de usuario 6	163

Tabla 54. Historia de usuario 7	163
Tabla 55. Historia de usuario 8	164
Tabla 56. Historia de usuario 9	164
Tabla 57. Historia de usuario 10	165
Tabla 58. Historia de usuario 11	166
Tabla 59. Historia de usuario 12	167
Tabla 60. Tarea de usuario 1	168
Tabla 61. Tarea de usuario 2	168
Tabla 62. Tarea de usuario 3	169
Tabla 63. Tarea de usuario 4	169
Tabla 64. Tarea de usuario 5	170
Tabla 65. Tarea de usuario 6	170
Tabla 66. Tarea de usuario 7	171
Tabla 67. Tarea de usuario 8	171
Tabla 68. Tarea de usuario 9	172
Tabla 69. Tarea de usuario 10	172
Tabla 70. Tarea de usuario 11	173
Tabla 71. Tarea de usuario 12	173
Tabla 72. Tarea de usuario 13	174
Tabla 73. Tarea de usuario 14	174
Tabla 74. Tarea de usuario 15	175
Tabla 75. Tarea de usuario 16	175
Tabla 76. Tarea de usuario 17	176
Tabla 77. Tarea de usuario 18	176
Tabla 78. Tarea de usuario 19	177
Tabla 79. Tarea de usuario 20	177
Tabla 80. Tarea de usuario 21	178

Tabla 81. Tarea de usuario 22	178
Tabla 82. Tarea de usuario 23	179
Tabla 83. Tarea de usuario 24	179
Tabla 84. Tarea de usuario 25	180
Tabla 85. Tarea de usuario 26	180
Tabla 86. Tarea de usuario 27	181
Tabla 87. Tarjeta CRC Box/Caja	181
Tabla 88. Tarjeta CRC Category/Categoría	182
Tabla 89. Tarjeta CRC Configuration/ Configuración	182
Tabla 90. Tarjeta CRC D estado de entregas	182
Tabla 91. Tarjeta CRC F Formas de pago	183
Tabla 92. Tarjeta CRC P Estado de los pagos	183
Tabla 93. Tarjeta CRC Message/ Mensaje	183
Tabla 94. Tarjeta CRC Operation / Operación	184
Tabla 95. Tarjeta CRC Operation Type/ Tipo de operación	184
Tabla 96. Tarjeta CRC Payment/ Pago	184
Tabla 97. Tarjeta CRC Spend/ Gasto	184
Tabla 98. Tarjeta CRC XX/ inicializar el gráfico	185
Tabla 99. Tarjeta CRC YY/ inicializar el gráfico	185
Tabla 100. Tarjeta CRC Saving/ Ahorro	185
Tabla 101. Tarjeta CRC Sell/ Facturas	185
Tabla 102. Tarjeta CRC Brand/ Marca	185
Tabla 103. Tarjeta CRC Person/Personas	186
Tabla 104. Tarjeta CRC Payment Type/ tipo de pago	186
Tabla 105. Tarjeta CRC User/Usuarios	186
Tabla 106. Tarjeta CRC Stock/ Existencias	186
Tabla 107. Tarjeta CRC Price/precio	187

Tabla 108. Tarjeta CRC Product/producto	187
Tabla 109. Historial de seguimiento de ejecución de pruebas	214
Tabla 110. Ingreso correcto al sistema	218
Tabla 111. Ingreso incorrecto al sistema	219
Tabla 112. Registro del nuevo producto	220
Tabla 113. Realización de venta	221
Tabla 114. Realización de compra	222
Tabla 115. Módulo de catálogos	223
Tabla 116. Módulo de herramientas	224
Tabla 117. Módulo de finanzas	225
Tabla 118. Módulo de inventario.	226
Tabla 119. Módulo de reportes.	227
Tabla 120. Módulo de administración.	228

ÍNDICE DE ANEXOS

Anexo 1. Certificado o acta del perfil de investigación	102
Anexo 2. Certificado del abstract por parte del centro de idiomas	103
Anexo 3. Certificado de plagio turnitin	105
Anexo 4. Autorización de recolección de información para la realización del proyecto	106
Anexo 5. Certificado de culminación del proyecto de investigación	107
Anexo 6. Entrevista a la señorita administradora y al señor bodeguero	108
Anexo 7. Tabla comparativa de metodologías ágiles	111
Anexo 8. Imagen de tabla comparativa de las arquitecturas de software	118
Anexo 9. Tabla comparativa de los lenguajes de programación	120
Anexo 10. Imágenes de tablas comparativas de los frameworks de desarrollo web	122
Anexo 11. Imagen de tabla comparativa de los gestores de bases de datos	146
Anexo 12. Plan de desarrollo de software	148
Anexo 13. Manual de usuario del sistema	229

RESUMEN

El objetivo de este proyecto, titulado “Propuesta de un Sistema de Gestión de Inventarios para la Farmacia del Hospital del Dia Nova Salud”, fue diseñar una herramienta tecnológica para el control de inventario. Durante la investigación se utilizó un enfoque cualitativo, con tipos de investigación descriptiva y exploratoria, que ayudaron a entender el problema. Para la recolección de datos se utilizó técnicas de campo, que implicó al departamento de farmacia. Con base en los resultados, se identificaron problemas en el manejo de datos, los registros que se estaban llevando causaban problemas, incluyendo pérdidas de materiales y accesorios, además de que solo se podían controlar hasta cierto punto. Como resultado, se obtenía un Kardex incorrecto, las herramientas ofimáticas no eran personalizables y generaban duplicación de información. Para el diseño del sistema se integraron tecnologías de desarrollo web, utilizando una arquitectura de Modelo vista Controlador, se combinó con un framework CodeIgniter v4 con el lenguaje PHP y base de datos MySQL. Estas herramientas juntas crean una sola entidad, las herramientas utilizadas se basaron en un marco teórico. El desarrollo del sistema permitió que el departamento de farmacia obtuviera una gestión en tiempo real al mismo tiempo que facilitaba el almacenamiento de documentos.

Palabras clave: Farmacia, Sistema, Gestión, Control, Inventario, Procesos.

ABSTRACT

The objective of the present project entitled “Proposal for the Inventory Management System for the Nova Salud Day Hospital Pharmacy”, was to design a technological tool for inventory control. Thus, during the investigation a qualitative approach based on descriptive and exploratory research was presented for the problem development. For data collection field techniques were used that involved the pharmacy department. Moreover, based on the results, problems were identified in data management since the records that were being kept were causing problems, as well as the loss of materials and accessories; besides, to the fact that it could only be controlled up to a certain point. As a result, an incorrect Kardex was obtained, the office tools were not customizable and generated duplication of information. Otherwise, for the design of the system web development technologies were integrated, using a Model View Controller architecture combined with a CodeIgniterv4 framework with the PHP language and MySQL database. Finally, the tools used allowed the development of the system that made the pharmacy department obtain real-time management while facilitating the storage of documents.

Keywords: Pharmacy, System, Management, Control, Inventory, Processes.

INTRODUCCIÓN

En la actualidad las organizaciones del área privada de salud, procesan a diario una gran cantidad de información de productos o servicios que ofertan sus establecimientos, partiendo de esta necesidad las empresas se enfocan en el desarrollo de plataformas tecnológicas que centralizan los datos de la organización para facilitar el acceso de información y la optimización de procesos que tienen dentro de su ciclo operativo como; control del inventario, rotación, registros, etc. El Hospital del Día Nova Salud de la ciudad de Tulcán, actualmente maneja diferentes procesos administrativos que están interconectados con diferentes departamentos, el tema principal de esta investigación recae en el control de inventario del departamento de farmacia, el cual es gestionado de forma manual dificultando la obtención de un control de inventario real, al ser tratados los datos para su debido uso se muestra que no son los que verdaderamente se encuentran en stock de farmacia, además existen pérdidas de materiales y accesorios dentro de su inventario, pues la forma en la que se registran los datos no es la más adecuada provocando múltiples retrasos. Ante ello, el estudio tiene como objetivo principal desarrollar un sistema informático, que permita realizar un adecuado control de inventario, de los materiales y accesorios dentro del stock que se desarrollan en el departamento de farmacia y trazar un marco teórico y metodológico que sirva como base para el desarrollo de este sistema, que integre la información que converge en esta área utilizando tecnología moderna que se adapte a las necesidades del establecimiento de salud. La importancia de esta investigación se fundamenta en la adquisición de conocimiento sobre procesos administrativos en una entidad privada y la sistematización de un flujo de trabajo completo, modelado en el desarrollo de esta investigación que centraliza las operaciones del departamento en un entorno escalable y seguro dentro de un sistema. El enfoque cualitativo de investigación permitió analizar y dimensionar las variables de estudio sobre el sistema de control de inventario. Se estableció la modalidad de campo, exploratoria, descriptiva, observación simple no regulada e investigación no experimental para recolectar información del departamento de farmacia, mediante la entrevista no estructurada se aplicó dos entrevistas dirigidas a la administradora y al bodeguero del establecimiento de salud. La construcción de la propuesta está completamente enfocada al desarrollo del sistema y fue guiada por los modelos de la metodología de software y la información recolectada con los instrumentos de investigación, dando lugar a la adaptación del módulo del control de inventario, las tecnologías utilizadas se centran en el lenguaje de programación PHP para así tener una correlación entre los componentes utilizados para su total desarrollo.

I. PROBLEMA

1.1. PLANTEAMIENTO DEL PROBLEMA

Actualmente la administración de la información en las organizaciones constituye el eje central de la parte operativa, la generación y estudio de datos permitió la adopción de novedosas maneras de idealización estratégica y financiera, esto es vital para las economías de muchos países en desarrollo, uno es Perú. Según el estudio realizado por Américaeconomía (2017), en la actualidad muchas organizaciones en Perú han pasado por un estudio nacional llamado victimización de empresas, echo por el Instituto Nacional de Estadística y Geografía (INEGI), en las cuales se realizó un estudio a grandes y pequeñas empresas llamadas PYMES, también se verificó que estas generaban alrededor de 4 millones de pérdidas por no mantener un adecuado control de inventario. A esta situación se suma que 70 de cada 100 PYMES permanecen en el área administrativa durante no más de 5 años.

El objetivo de un control de inventario es, mantener todos los materiales del establecimiento en orden, para tener una mayor fluidez de información cuando ésta sea requerida en su debido tiempo y así poder mantener un equilibrio constante y duradero.

En Guayaquil-Ecuador se realizó un estudio enfocado al control de inventario, la motivación que dio origen a la investigación fue la pérdida generada en años pasados, durante la investigación se observó que muchos de los problemas partían del control de su inventario, pues este no cumplía con las medidas y el uso de los recursos tecnológicos adecuados para realizar actividades. Se encontró que, debido a la transición del stock, no era factible llevar un estricto control en el departamento de bodega. Se requería establecer parámetros y la elaboración de cuadros informativos constantes de los movimientos de los productos, es decir, empezar a tener un conteo manual y generar un Kardex por ítem, por esta razón se sugirió que la compañía incorporará un sistema que proporcione un módulo de inventario eficiente, además de permitir realizar los reportes necesarios, se sugirió mediante la elaboración de diagramas de flujo la reorganización de los procesos más críticos de la logística que se manejaban en el lugar con la finalidad de mejorarlos. (Cornejo Gómez, 2017)

Una situación similar se observó a nivel local, en el Hospital del Dia Nova Salud de Tulcán, donde se abordan de manera más detallada los problemas de gestión del inventario, una de las cuales está relacionada con el control de stock de farmacia. se ha presentado una problemática en este departamento por la inadecuada gestión del inventario, teniendo pérdidas de materiales y accesorios que conforman el stock , al igual que solo se podía controlar hasta un límite determinado, Convirtiéndose en un problema de gran relevancia para el Hospital, llegando a perder el valor real que existe en stock que se maneja en farmacia, teniendo un valor inadecuado de la ganancia o pérdida del inventario, además de perder el control directo de los medicamentos que están por caducar y cuantos se encontraban disponibles en stock, al no verse apoyados en un sistema que les ayude a manejar un adecuado control del inventario se llegó a un punto crítico que obligó a los trabajadores del área a realizar el conteo manualmente generando un Kardex erróneo, transformando así, a las herramientas que se utilizó en su momento, en inútiles e inservibles, este problema sigue vigente en la actualidad, causando más pérdidas económicas y necesidades para el establecimiento de salud. (Carmen Zúñiga, comunicación personal, 02 de septiembre, 2019)

1.2. FORMULACIÓN DEL PROBLEMA

¿Qué efecto tendría la aplicación de un sistema de Inventario en el control de recursos que se utilizan en el proceso de compra y uso en la farmacia, del Hospital del Día Nova Salud de la ciudad de Tulcán en el periodo 2021?

1.3. JUSTIFICACIÓN

Dentro de una organización empresarial se debe controlar todos los niveles, además, considerar que esta es una inversión significativa, misma que, si no se maneja de forma adecuada afectaría la gestión financiera de la empresa como lo mencionan el estudio generado por Américaeconomía (2017).

La investigación ayudará a comprender los procesos en base a la teoría donde se verá apoyada para solucionar el problema al cual se enfrenta desde hace varios años el Hospital del Día Nova Salud, esta necesidad no estará totalmente cubierta, reducirá el problema y evitará la pérdida de materiales y accesorios del inventario en farmacia.

El beneficiario directo del proyecto es el departamento de Farmacia, permitiendo recolectar datos importantes por medio de los trabajadores relacionados a los procesos de la gestión del inventario que se manejan actualmente, además, el campo en el cual se desarrolla la investigación permite estudiar procesos de gestión dentro de una institución privada, logrando la recolección de datos que sirva para desarrollar una solución técnica donde se pueda proporcionar una herramienta tecnológica donde se brinde información real de los material y accesorio en el inventario, a esto se suma que se puedan identificar con prontitud los medicamentos que están a punto de caducar y puedan hacer la respectiva devolución al proveedor, antes que se genere pérdidas para el establecimiento, finalmente, el trabajo de investigación es factible y fiable al tener información accesible, el apoyo por parte de las autoridades del establecimiento de salud.

1.4. OBJETIVOS Y PREGUNTAS DE INVESTIGACIÓN

1.4.1. Objetivo General

Diseñar una herramienta tecnológica para el control del inventario en el Hospital del Día Nova Salud.

1.4.2. Objetivos Específicos

- Identificar los procesos con falencias que tiene el Hospital del Día Nova Salud para el control del inventario.
- Analizar los requerimientos de sistematización de materiales y accesorios de stock en farmacia para el control de inventario.
- Desarrollar un sistema informático, que permita realizar un adecuado control de inventario, de materiales y accesorios dentro del stock de farmacia del Hospital del Día Nova Salud.

1.4.3. Preguntas de Investigación

- ¿De qué forma se Identificará los procesos con falencias que tiene el Hospital del Día Nova Salud para el control del inventario?
- ¿De qué forma se analizarán los requerimientos de sistematización de materiales y accesorios de stock en farmacia para el control de inventario?
- ¿De qué forma se desarrollará el sistema informático para el control del inventario en el Hospital del Día Nova Salud?

II. FUNDAMENTACIÓN TEÓRICA

2.1. ANTECEDENTES INVESTIGATIVOS

Para reforzar el estudio de las variables de investigación se han recopilado antecedentes con mayor relevancia, estos trabajos fueron extraídos de repositorios digitales de instituciones de educación superior.

En el año 2018 se desarrolló una propuesta de aplicación web basada en control de inventarios para la empresa “Dgc Jeans” en Riobamba, Ecuador, por Sánchez Villalva (2018), en la Escuela Superior Politécnica de Chimborazo, el cual se concentró en las áreas de; compra, venta, gestión y control del inventario, estas áreas eran de mayor dificultad en cuanto a control, ante ello se desarrolló un sistema web en el cual se utilizó la metodología SCRUM, Los componentes que hicieron posible este desarrollo y administración fueron Oracle 11g XE, CISBlue, junto con el uso de Framework Spring MVC con su uso de bibliotecas, primease y Bootstrap, que ayudaron en el desarrollo general del sistema, para finalizar la investigación se presentó la siguiente conclusión: Se realizó un análisis del funcionamiento del sistema y de los procesos en los departamentos de bodega, producción y venta-traspaso. El análisis permitió identificar las deficiencias encontradas en estos departamentos, debido a la falta de un control adecuado de todos sus procesos, la implementación de un nuevo sistema dio un porcentaje del 83.6 % de adaptabilidad y funcionalidad.

En el año 2018, Lema Sagbaycela de la Escuela Superior Politécnica de Chimborazo diseñó una aplicación web para el control de inventario y sus áreas en la empresa Aluminio y Vidrio “López” en Riobamba, Ecuador, A pesar de tener un enfoque de control, ventas, facturación y relaciones públicas, esta empresa carecía de un sistema que pudiera administrar estos procesos y permitiera tener un inventario en tiempo real de todos los materiales que se encontraban en la tienda, lo que generaba una pérdida de tiempo en el manejo de inventarios. Ante esta situación, se decidió construir un nuevo sistema utilizando la metodología Lean Software Development en conjunto con MYSQL como base de datos, así como un entorno de desarrollo integrado (IDE) y un JSP para la lógica de negocios. Todos estos elementos fueron esenciales para la construcción del sitio web con el fin de lograr un rendimiento óptimo en el uso diario, utilizando los resultados de este estudio, se descubrió que al aplicar el software se obtienen los siguientes valores en términos de porcentajes: el primer valor fue de 86 % de aceptabilidad al ser aplicado el sistema web, el segundo fue del 92 % de satisfacción al no tener pérdida de información en su totalidad.

En Quito, Ecuador, en el año 2018, Vinueza Aizaga, de la Universidad Tecnológica Israel, diseñó un sistema de información para rastrear la existencia de una farmacia central para el establecimiento Farmacia el Descuento, esta empresa no llevaba un registro real dentro de bodega, esto daba como consecuencia datos erróneos al gestionarla, por esta razón el desarrollo se enfocó en mejorar los procesos de, control, ventas, facturación y publicidad, asistiendo en la mejora de los controles de entrada, salida y búsqueda de productos. Para desarrollar el sistema, el autor utilizó la metodología Xp, agregando la arquitectura MVC (Modelo Vista Controlador). Este modelo se utiliza para aplicaciones basadas en la simplificación de la interfaz de usuario conocida como JSF (Java Server Faces), y para la agilidad del sistema, el autor desarrolló una base de datos POSTGRE. Teniendo todo esto en cuenta, el autor llegó a la siguiente conclusión, al aplicar el sistema como nueva herramienta en las sucursales de farmacia, proporcionaba datos y cantidades reales dentro de stock, llevando un mayor control en bodega central y en la gestión de compras y abastecimiento.

Con los antecedentes de investigación antes mencionados, se opta por desarrollar un sistema web, por la escalabilidad que estos poseen dentro de un entorno de trabajo. Las investigaciones se apoyaron de metodologías ágiles como, Scrum, Lean Software Development y por último Xp, al analizar los resultados que estas investigaciones obtuvieron según el caso de estudio al que se aplicó, llevándolas a un mismo objetivo que era, optimizar procesos para lograr controlar un inventario.

2.2. MARCO TEÓRICO

2.2.1. Control de inventario

El control de inventario se define como "un punto determinante en el manejo estratégico de toda organización con relación a las tareas correspondientes al control del inventario, se relaciona con la determinación de los métodos de registro y puntos de rotación" (Jaile, 2018). Al comprender el concepto sobre el control del inventario se tiene una idea clara sobre el tema que estamos abordando dentro de la investigación, debemos conocer a dónde queremos llegar y hasta dónde podemos expandir el tema, debe comprenderse que es muy extenso, pero podemos delimitarla con las bases adecuadas.

2.2.1.1. El objetivo del Control del Inventario

La administración de un inventario se ocupa principalmente de los procesos y productos terminados. Este control vendrá determinado por el tipo de establecimiento y su estructura organizativa. El objetivo principal del control de inventario es funcionar como un regulador entre las tasas de absorción y las cadencias o el consumo de sus salidas. (Quizhpi, 2018)

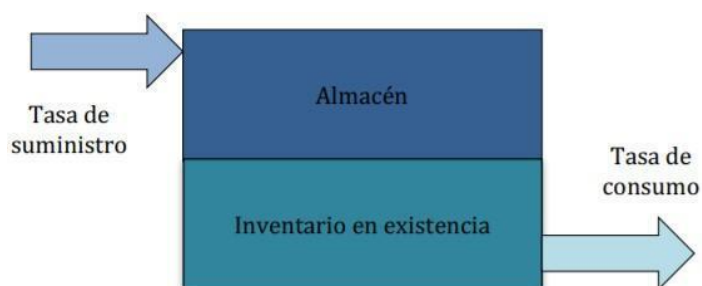


Figura 1. Modelo de control de inventario

Fuente: Modelos de Inventario 2017 por Pedro Olave

El objetivo principal del control de inventario es organizar productos específicos en el momento adecuado, debe comprenderse que tanto la gestión como el control tienen como objetivo el actuar como medio de regulación, es muy importante para los procesos que se especifican dentro de una organización.

2.2.1.2. Ventajas de un sistema de control

El autor Culqui (2019) define, "las ventajas de un sistema sobre el control de inventario, significa adoptar procedimientos organizativos que permitan".

- Proporcionar toda la información relativa a los materiales y anexos con el fin de mantener la gestión del inventario.
- Mantener un inventario adecuado de los artículos en el inventario.

- Obtener un archivo histórico.
- Anticipa y satisface la demanda futura mientras reduce el nivel de riesgo.
- Teniendo en cuenta las condiciones y restricciones del proveedor, así como la reducción de costes de gestión.
- Tener una reserva dentro del stock.

Al tomar en cuenta los ítems establecidos, se logra tener una idea clara de cómo se debe llevar un adecuado procedimiento en el área de control y almacenamiento, misma que sirve en el análisis de los procesos que lleva el establecimiento.

2.2.1.3. Tipos de procesos:

- El método de ensamblaje de la fabricación puede determinar la calidad de cada producto que se fabrica o almacena.
- Los pedidos especificados se cumplen inmediatamente después de completar los materiales primarios, después de recibir el pedido exacto del producto.
- Los materiales para la producción continua se compran con anticipación y los productos terminados se mantienen en stock por un corto período de tiempo.

Estos tipos de procesos que mantienen una regla de orientación y de organización son esenciales al momento de analizar nuevos procesos de almacenamiento, ya sea para su venta o para ser almacenados permitiendo llevar un adecuado control.

2.2.2. Tipos de inventario

2.2.2.1. Inventario de materias primas o insumos

Los elementos utilizados en la creación de un producto que han sido transformados e integrados en el resultado final. Las materias primas se utilizan principalmente en empresas industriales, como las que fabrican bienes. Para determinar el costo final de un producto y sus ingredientes, debe ser completamente identificable y medible. (Cardona y Rojas, 2018)

2.2.2.2. Inventarios de materiales semiacabados o productos en curso

“Los productos semiacabados también se pueden utilizar para describir un inventario de productos en proceso. Estas son materias primas que han sido rescatadas de un almacén de materias primas y ahora se están transformando en productos terminados” (Alava, 2021).

2.2.2.3. Inventario de productos terminados

Según el autor Serna (2018) dice que, cuando se habla de este tipo de inventario se refiere a una empresa, una fábrica, más específicamente suele utilizar diferentes bienes para fabricar sus productos finales, cuando el producto se almacena correctamente, ya no necesita pasar por la línea de producción y se puede vender, se puede decir que es un inventario de producto terminado.

2.2.2.4. Inventario de materiales de empaque y embalaje

El autor Paredes (2018) menciona que, en lo que respecta a los sistemas de embalaje e inventario, el almacenamiento incluye todas las actividades necesarias para almacenar y mantener los productos, desde la fabricación o adquisición del producto hasta la venta. Es necesario ajustar y compensar la oferta y la demanda. Significa la suficiencia entre el número de transacciones.

Para tener un producto terminado este transcurre por varios procesos, desde su combinación o fabricación, hasta su fase de elaboración final, una vez allí, el producto pasa a un área de orden en la cual forma parte de un stock específico para ser distribuido, esto ayuda a mantener claro el concepto de cómo un producto debe ser ordenado y caracterizado a la vez.

2.2.3. Clasificación concepción logística

2.2.3.1. Inventarios de lote o de ciclos

Estas son las cosas que se requieren para respaldar las decisiones tomadas en función del tamaño del lote. Esto ocurre cuando se toma la decisión de trabajar en lotes en lugar de comprar, producir o enviar una sola unidad a la vez; como resultado, el inventario tiende a acumularse en varios lugares del sistema; esta es la idea de administrar un inventario sólido. Disponer de lotes individuales de productos que permite un mayor control sobre la totalidad de los mismos, así como la capacidad de llevar un pedido y una existencia real para la distribución. (Cabello, 2019)

2.2.3.2. Inventarios de seguridad

El autor López (2018) menciona que, es un activo que existe en las empresas debido a la oferta o demanda unitaria. El stock de seguridad de materias primas puede evitar que el desempeño del proveedor sea incierto debido a factores como el tiempo de espera, huelgas y vacaciones, se utiliza para prevenir la escasez causada por fluctuaciones inciertas.

Tabla 1.
Clasificación de inventarios ABC

A	B	C
Productos de alto valor, de gran venta, que requieren de mayor atención y cuidado a través de:	Productos de alto valor con ventas moderadas, requiere un tratamiento normal; es decir una atención ajustada a los requerimientos del negocio	Productos de bajo valor y poca venta, que deben tratarse según el principio de la simplificación productiva y administrativa y de reducción de costos.
Análisis de mercado, de precios y de costos		Requerimientos simplificados de inventarios
Registro y control de inventarios		Trámites simplificados en el manejo de pedidos y pedidos de grandes cantidades
Determinación precisa de seguridad		Supervisión de las existencias
Aplicación preferencial del análisis de valores		

Fuente. Ramírez (2019). Clasificación de Inventarios ABC

Los siguientes factores deben ser considerados al determinar la importancia de un producto:

- Ventas anuales.
- Costo unitario.
- Oferta y demanda de materiales.
- Disponibilidad de recursos de fabricación.
- Confiabilidad del proveedor.

2.2.4. Clasificación impulsada por la demanda

Haciendo referencia a lo que menciona el autor González (2020), el inventario se divide en dos categorías: demanda dependiente y demanda independiente:

- La demanda independiente es determinada directamente por el mercado.
- Repuestos requeridos por los clientes.
- La demanda dependiente o requisitos son relacionados con otro artículo.

La demanda es parte de uno de los dos términos conocidos como la “ley de la oferta y la demanda”. Según esta ley, el sistema de distribución de precios de un mercado en particular está regido por productores que brindan bienes y servicios y consumidores que desean obtenerlos.

2.2.5. Tipos de modelos de inventarios

Según los autores Batero y Castro (2018) dicen que, existen tres tipos de modelos de inventario: modelo P, modelo Q y sistema de control. Estos modelos son los encargados de organizar, determinar y mantener actualizada la información requerida por los distintos departamentos. Son un conjunto de normas, políticas y procedimientos que se aplican al mantenimiento y control de activos intangibles tales como materiales y productos que se utilizan en una organización, a la vez que permiten el desarrollo de operaciones posteriores:

- Organizar pedidos y recibos.
- Determinar el tamaño del pedido y el tiempo que llevará completarlo.
- Realizar un seguimiento de la información del pedido en todo momento.

2.2.5.1. Organizar pedidos y recibidos "Sistema P"

Un sistema de período constante, también conocido como sistema periódico, establece un ciclo constante entre cada pedido, cuyo tamaño está determinado por los niveles de inventario y la demanda anticipada.

2.2.5.2. Determinar el tamaño de la solicitud y cuándo se completará "Sistema Q"

La característica principal del sistema es que puede ejecutar lotes del mismo tamaño según sea necesario, según los niveles de inventario y la necesidad de almacenamiento anticipada.

2.2.5.3. Mantener actualizada la información del pedido en todo momento "Sistema de Control"

Todos los sistemas de gestión de inventario incluyen un sistema de control capaz de realizar las siguientes funciones:

- Mantener un registro de las actualizaciones de inventario.
- Notifique los niveles de inventario para que sepa cuándo ordenar y cuánto ordenar de cada producto.
- Condiciones de información mediante indicadores de falla del sistema.
- Creación de informes para la gestión y los administradores de inventario.

Lo que mencionan los autores son los tres tipos de sistema de inventario que se puede utilizar dentro de un modelo de control, con estas ideas podremos modelar fases de diseño de un software para mantener un estricto control de un inventario en específico.

2.2.6. La gestión del inventario

De acuerdo con el autor Hernández (2017), un sistema de control de inventarios es un proceso que influye significativamente en todos los aspectos de las operaciones de una organización, además es parte fundamental de la gestión, si no se controla se traduce en mayores costos. y requiere más inversión. Tener un sistema de control de inventario tiene una serie de ventajas, entre ellas:

- Mejora la calidad del servicio al cliente, reduce las pérdidas de ventas debido a la escasez de inventario y aumenta la lealtad del cliente hacia la empresa.
- Mejorar el flujo de efectivo de la empresa al tener compras más eficientes y una mayor rotación de inventario garantizará que el dinero no se quede ocioso en el banco, sino que se utilice.
- Poseer la capacidad de determinar el estado de un producto, por lo tanto, ayudar en la planificación del inventario.
- Encontrar elementos estacionarios o de movimiento lento para un diseño estratégico que sea fácil de deconstruir.
- Reduce los costos de envío para planificar mejor y reducir las compras de emergencia.
- Supervisar la calidad del producto identificado y asegurarse de que se supervisa correctamente en términos de calidad del producto.
- Incrementar la rentabilidad por metro cuadrado liberando y optimizando el espacio de almacenaje.

Tener un adecuado manejo de inventario es esencial, permite deducir el producto que logró tener mayor o menor acogida, al igual que productos caducados, se logrará además devolverlo sin generar pérdidas.

2.2.7. Definición de stock

Dentro del stock se encuentran productos con alta demanda por parte del consumidor final, así como aquellos que están a la espera de ser ingresados a un nuevo proceso de inventario. Según Méndez (2018), “stock se refiere a cualquier artículo que se almacena con el propósito de venderlo o utilizarlo en un proceso de fabricación, y existen varios tipos de stock que entran en esta categoría”, el autor menciona los tipos de stock que existen.

2.2.7.1. Stock óptimo

Es la cantidad de materiales y accesorios que permite a una farmacia cubrir su demanda con la menor inversión.

2.2.7.2. Stock de seguridad

Es el menor número de unidades de producto que permite prestar un servicio mientras se procesa otro pedido.

2.2.7.3 Stock circulante

Es la cantidad o el pedido pre planificado en las compras a proveedores.

2.2.7.4. Stock ciego

Estos son los artículos que se guardan en una tienda o almacén.

2.2.7.5. Stock vivo

Estos son los productos o materiales que están a disposición de los pacientes o clientes.

2.2.7.6. Stock ideal

Un inventario óptimo nos permite satisfacer las necesidades de nuestros clientes con la menor inversión. Esto está determinado por una serie de factores, que incluyen:

- Compra de productos de alta demanda.
- El número de compras está determinado por el tiempo que un artículo está disponible en el mercado.
- Comprar con excesiva precaución puede resultar en una interrupción de las demandas, lo que puede dañar la economía y la reputación.

Al tener la idea de que existen varios tipos de stock, estos se adaptan a la necesidad específica que se puede tener en su momento, se puede elegir uno de ellos según la demanda.

2.2.8. Métodos de valoración de inventario

“El método de valoración de inventario es una técnica para determinar el valor de las unidades de inventario en términos de dinero. Este tipo de valoración es especialmente útil cuando las unidades se compran a precios diferentes” (Ormaza, 2020), el autor menciona los tipos de métodos que existen.

2.2.8.1. Método FIFO o PEPS

Este método se basa en el principio de primero entrar y luego salir. Debido a que se basa en los costos actuales, su evaluación es más apropiada para las condiciones del mercado.

2.2.8.2. Método LIFO O UEPS

Teniendo en cuenta que todos los productos básicos que quedaron en último lugar son los que salen primero. Su ventaja se deriva del hecho de que el inventario mantiene un valor constante cuando los precios suben.

2.2.8.3. Método de costo promedio aritmético

El resultado estará determinado por la media aritmética del precio unitario del producto.

2.2.8.4. Método del promedio armónico o ponderado

Este promedio se puede calcular multiplicando el precio por el número de unidades compradas y luego dividiendo la suma por el número de unidades compradas.

2.2.8.5. Método del costo promedio móvil o del saldo

Calcule el valor de los activos en función de los cambios de entrada y salida (compra o venta) para proporcionar un promedio continuo.

2.2.8.6. Método del costo básico

Este método asigna un valor fijo al inventario más pequeño. Es bastante similar a LIFO, excepto que solo se aplica al inventario más pequeño.

2.2.8.7. Método del precio de venta al detal

Le permite estimar el inventario con la frecuencia requerida. El inventario físico se realizará de acuerdo al precio de venta indicado en el material.

2.2.8.8. Costo de mercado o el más bajo

Se tomó como base el precio más bajo de los activos, con principios de conservación confiables que no anticiparon ganancias ni previeron pérdidas potenciales.

2.2.9 Software de control de inventario

El software es una herramienta que facilita el entorno dentro de una gestión, pues es intermediario entre el usuario y los productos almacenados, se utiliza para llevar un registro real de un stock.

2.2.9.1. Tipos de software de manejo de inventario

El sistema de control de inventario es una solución técnica que integra todos los aspectos de las tareas de inventario de la organización, incluido transporte, compra, recepción, almacenamiento, facturación, seguimiento y reordenamiento, la verdad es que un buen sistema de control de inventario logra todo esto, adoptando un enfoque distinto del inventario y permitiendo que la organización utilice procedimientos adecuados, esto es lo que menciona el autor (Alejos, 2019), además indica que existen 4 tipos de sistemas de manejo de inventario, estos son:

2.2.9.2. Software de inventario perpetuo

Un sistema de inventario perpetuo, actualizará continuamente los registros de inventario y calculará aumentos y disminuciones a medida que recibe artículos de inventario al manejar más registros que un sistema de inventario periódico. Algunas organizaciones prefieren los sistemas de inventario perpetuo porque brindan información de inventario actualizada y manejan mejor los recuentos mínimos de inventario físico, (Cerquera, 2019) el autor también menciona sus características.

- Registrar los movimientos después de cada transacción.
- Se registra la venta para realizar una entrada para disminuir el inventario y aumentar el COGS “Cost of Goods Sold por sus siglas en inglés- Costo de los bienes vendidos”.
- COGS refleja los saldos actuales en cualquier momento.
- Muestran detalles más específicos.
- Por cada registro nuevo se registra un nuevo libro.
- A medida que crecen son más informatizados.
- Requieren más registros a comparación de los otros.

2.2.9.3. Software de inventario periódico

Los sistemas de inventario periódico no rastrean el inventario diariamente; en cambio, permiten que las organizaciones comprendan los niveles de inventario inicial y final durante un cierto período de tiempo. Estos tipos de sistemas de control de inventario utilizan recuentos de inventario físico para realizar un seguimiento del inventario. Una vez que se complete el inventario físico, el dinero en la cuenta corriente se transferirá a la cuenta y se ajustará para que corresponda al costo del inventario al final del período. (Muñoz, 2021), continúe leyendo para ver las características que menciona el autor:

- Al realizar un cálculo se debe recontar manualmente.
- La mayoría de las cuentas son compras “reabastecimientos”.
- El Kardex que se maneja es únicamente datos en cantidades.
- Se lleva un registro y control permanente.
- Información sobre la cantidad y disponibilidad del inventario.
- Brinda información de sus ventas en cualquier momento.
- El costo de ventas se actualiza automáticamente cuando se lo cambia.

2.2.9.4. Software de inventario de código de barras

Los sistemas que utilizan tecnología de código de barras para el control de inventario son más precisos y eficientes que los que utilizan procesos humanos. Cuando se usa como parte de un sistema de control de inventario general, el sistema de código de barras actualiza el nivel de inventario automáticamente cuando los empleados lo escanean con lectores de códigos de barras, escáneres o dispositivos móviles. Las siguientes son las ventajas que el autor (Marshall, 2020) menciona al usar códigos de barras:

- Registro de todas las transacciones del inventario.
- Eliminación de datos erróneos que ocurren al hacer registros manuales.
- Eliminación de datos erróneos de entrada.
- Reactualización de inventario del sistema por completo.
- Facilidad de búsqueda de los registros.
- Facilidad de movimiento entre inventarios.

2.2.9.5. Software de inventario de Identificación por Radiofrecuencia (RFID)

Los sistemas de gestión de inventario RFID emplean tecnologías activas y pasivas. La tecnología RFID activa emplea lectores RFID fijos en todo el almacén; Las etiquetas RFID se pasan al lector y el movimiento se registra en el software, como resultado, los sistemas activos son mejores para las organizaciones que requieren seguimiento de inventario en tiempo real. La tecnología pasiva, por otro lado, requiere el uso de lectores portátiles para rastrear el movimiento del inventario. Los datos son registrados por el software de control de inventario cuando se lee la etiqueta. El rango de lectura de la tecnología RFID es de alrededor de 40 pies para la tecnología pasiva y de aproximadamente 300 pies para la tecnología activa, según los autores. (Amsler, 2021)

Existen algunos desafíos asociados con los sistemas de control de inventario RFID.

- Las etiquetas RFID son significativamente más caras que las etiquetas de código de barras.
- Se utilizan para productos de alto valor.
- Cuando las etiquetas RFID se utilizan en entornos metálicos o líquidos, pueden causar interferencias.
- El costo de cambiar a un equipo RFID también es alto.
- Tanto los clientes como las empresas de transporte requieren el mismo equipo.
- Las etiquetas RFID contienen más datos que las etiquetas de código de barras, lo que implica que el sistema y el servidor pueden sobrecargarse de información.

Por tanto, existen varios sistemas para el control de inventarios. Para elegir algún sistema que ayude a nuestro control diario se deberá conocer a detalle los ingresos y egresos de bodega o almacén, con la finalidad de identificar la viabilidad del sistema, la organización buscará obtener una solución completa que permita comprender el inventario, mejorando la precisión del valor real del seguimiento de los registro e informes que se utilizan dentro de una organización.

2.2.10. La Ingeniería del software en el control de inventario

Esta ingeniería abarca procesos dentro de áreas específicas como, controlar una pequeña bodega hasta un almacén de contenedores de distintas clases, al administrar un tipo de los que fueron nombrados es necesario llevarlo de la manera más adecuada como apoyarse en una herramienta que ayude a la gestión en las tareas desde, órdenes de compra hasta costes operativos, hasta llegar a la satisfacción de los clientes, el autor Barlot (2019) menciona lo siguiente:

Al saber que existen procedimientos que son de suma importancia dentro de una empresa o establecimiento debemos saber que se corre un riesgo para la mercadería que está por llegar o por venderse al consumidor final, afortunadamente existen software para la optimización de procesos basadas en la ingeniería de software, un software de control de inventarios realiza un seguimiento de pedidos, stock, entregas y ventas. Este ayuda a mantener un adecuado control de cantidades reales, además de administrar toda la información en tiempo real, evitando así el exceso de información inadecuada y de haber substock.

Se ha visto que la ingeniería de software consta de varias ramas, pero esto dependerá de lo que se pretenda controlar dentro de un almacén o bodega, esta ingeniería busca lograr un gran impacto dentro de la industria logística, donde los procesos sean llevados dentro de un software que ayude a mantener un adecuado orden de la información constante.

2.2.11. Metodologías de desarrollo de software

Según Maida y Pacienza (2017), las metodologías de desarrollo son una colección de herramientas, procesos, métodos y documentos de apoyo que ayudan a los desarrolladores de software en su búsqueda para crear nuevos sistemas de información.

Las metodologías se basan en una combinación de los modelos de proceso genéricos. Definen artefactos, roles y actividades, junto con prácticas y técnicas recomendadas. La metodología para el desarrollo de software es un modo sistemático de realizar, gestionar y administrar un proyecto para llevarlo a cabo con altas posibilidades de éxito. Una metodología de desarrollo de software abarca los procesos que deben seguirse de manera sistemática para conceptualizar, construir y mantener un producto de software desde el momento en que se necesita hasta que alcanza el objetivo para el que fue creado.

Beneficios Metodologías Ágiles

Calidad

Realizando pruebas desde el principio e iterando sobre el producto tras recibir el feedback.

Resultados

Entregando algo tangible y que aporte valor desde la primera iteración.

Flexibilidad

Permitiendo cambios de alcance, estimando y planificando de manera ágil.

Mantenibilidad

Creando un software de calidad, con casos de prueba y una documentación asumible.

Eliminación de riesgos

Validando cada entrega en sprints cortos y asegurando la calidad con casos de pruebas.

Motivación

Trabajando de manera conjunta con el cliente, viendo crecer el producto final tras cada iteración.

Figura 2. Beneficio de las metodologías ágiles
Fuente. Beneficio de las metodologías ágiles 2017 por Maida y Pacienza

2.2.11.1. Metodología Adaptive Software Development

“Es una metodología de implementación de patrones ágiles para desarrollo de software, el funcionamiento que tiene es de ciclo además de reconocer que en cada iteración se producirán cambios e incluso errores” (Simon, 2020). El autor menciona sus características.

- Iterativo.
- Orientado a los componentes del software.
- Tolera los cambios.
- Guiado por riesgos.
- Las revisiones sirven para comprender los errores y volver al ciclo de desarrollo.

2.2.11.2. Metodología Scrum

Scrum es un proceso en el que se aplica de manera regular en un conjunto de buenas prácticas, es aplicado de forma regular para trabajar en equipo y obtener el mejor resultado posible de un proyecto. Estas prácticas se complementan y su selección se basa en un estudio de cómo trabajar con equipos altamente productivos. En scrum se realizan entregas parciales y frecuentes del producto final, priorizando el beneficio al receptor del proyecto. (Tymkiw y Tumino, 2020) Algunas de sus características principales son las que mencionan los autores (González y Derby, 2019),

- Scrum maneja un modelo referente a un conjunto de prácticas y roles.
- Manejar regularmente las expectativas de los clientes.
- Resultados previstos.
- Adaptabilidad y flexibilidad.
- Tienen una buena mitigación de riesgos.

2.2.11.3. Metodología XP

Calvo (2018) menciona que cuando se trata de metodologías ágiles, hay muchas metodologías de desarrollo para elegir, incluida XP.

La metodología XP o Programación Extrema es una metodología ágil y flexible utilizada para la gestión de proyectos Extreme Programming, se centra en potenciar las relaciones interpersonales del equipo de desarrollo como clave del éxito mediante el trabajo en equipo, el aprendizaje continuo y el buen clima de trabajo. Esta metodología pone el énfasis en la retroalimentación continua entre cliente y el equipo de desarrollo, es idónea para proyectos con requisitos imprecisos y muy cambiantes.

A lo largo de la vida de una pieza de software, pueden ocurrir varias modificaciones, ya sea durante el desarrollo o después de que se haya completado. Se considera un método de edición y modificación en cualquier etapa de desarrollo, y se denomina programación extrema o XP.

Tabla 2.
Comparación de metodologías de desarrollo

Metodologías de desarrollo			
Características	Scrum	XP	ASD
Tamaño del proyecto Pequeños y medianos	✓	✓	✗
Modelo de desarrollo iterativo y rápido	✓	✓	✓
Desarrollo orientado a cliente	✗	✓	✓
Implementación de equipo pequeños	✓	✓	✗
Búsqueda de excelencia técnica	✓	✓	✓
Pruebas TDD	✓	✓	✓
Distribución física	-	✓	✓

Nota: La tabla 2 muestra características entre las metodologías ágiles, recopilación de varias fuentes, véase el Anexo 7.

En la tabla comparativa se establece que la Metodología de desarrollo Xp, es una de las más adecuadas en la elaboración de este proyecto, como entendemos la metodología se utiliza principalmente en proyectos de desarrollo de software de una sola persona. Con estas bases se puede construir un software adaptado a las especificaciones de cualquier cliente. Algunas pueden cambiar a lo largo del proceso de desarrollo. Por tanto, este método permite responder de forma flexible a los contratiempos y cambios, a continuación, véase la arquitectura en su totalidad.

Según los autores Canive y Balet, las ventajas de la metodología XP son: (2020).

- Tiene un horario bien organizado.
- Mejorar la eficiencia del proceso de planificación y pruebas.
- La tasa de errores es extremadamente baja.
- Alienta a los programadores a trabajar en un entorno positivo.
- Mejora la comunicación entre clientes y desarrolladores.
- Demostrar el cambio.
- Puede ser utilizado en cualquier entorno de programación.
- Pruebas continuas durante todo el proyecto.
- Esta metodología se utiliza en la implementación de nuevas tecnologías.

Las siguientes son las desventajas de la metodología XP.

- Se recomienda su uso en proyectos a corto plazo.
- La necesidad de adherirse estrictamente a los principios Xp.
- Puede que no siempre sea fácil, y la mejor opción es la elaboración tradicional.

2.2.11.4. Lista del equipo XP

Según la autora Saavedra (2021) hace mención a lo siguiente.

- **El cliente y su responsabilidad**

Los clientes son los encargados de definir los objetivos del proyecto, así como de supervisar su gestión. Haga una lista de los requisitos y prioridades del proyecto.

- **Programadores y su marco de acción**

Como expertos en actividades que ayudan a las personas a alcanzar sus objetivos, los programadores serán los encargados de definir la duración y estimar el tiempo necesario para completar el proyecto. Como resultado, planificarán el proyecto de acuerdo con los requisitos del cliente.

- **Testers y su rol en la metodología XP**

El Tester, o el responsable de las pruebas, amplía su ámbito de trabajo ya que la comunicación con el cliente es fundamental para alinear los resultados con los requisitos estimados.

- **Tracker o responsable del seguimiento**

Su objetivo es mantener el control y la razón de realización en todo momento, así como la comunicación y relación permanente con los clientes.

- **Entrenador y su responsabilidad**

El jefe de proyecto realiza una labor básica: asesorar y orientar permanentemente al equipo de trabajo y a los clientes. Sirven como una guía para el proyecto, asegurando que todos sepan qué hacer, cómo hacerlo y cuándo hacerlo.

- **El manager Xp y su respuesta en el método**

Responsable de coordinar las comunicaciones entre todas las partes, proporcionar y administrar los recursos necesarios y mantenerse actualizado sobre la funcionalidad y el estado del proyecto.

2.2.11.5. Metodología XP y sus fases

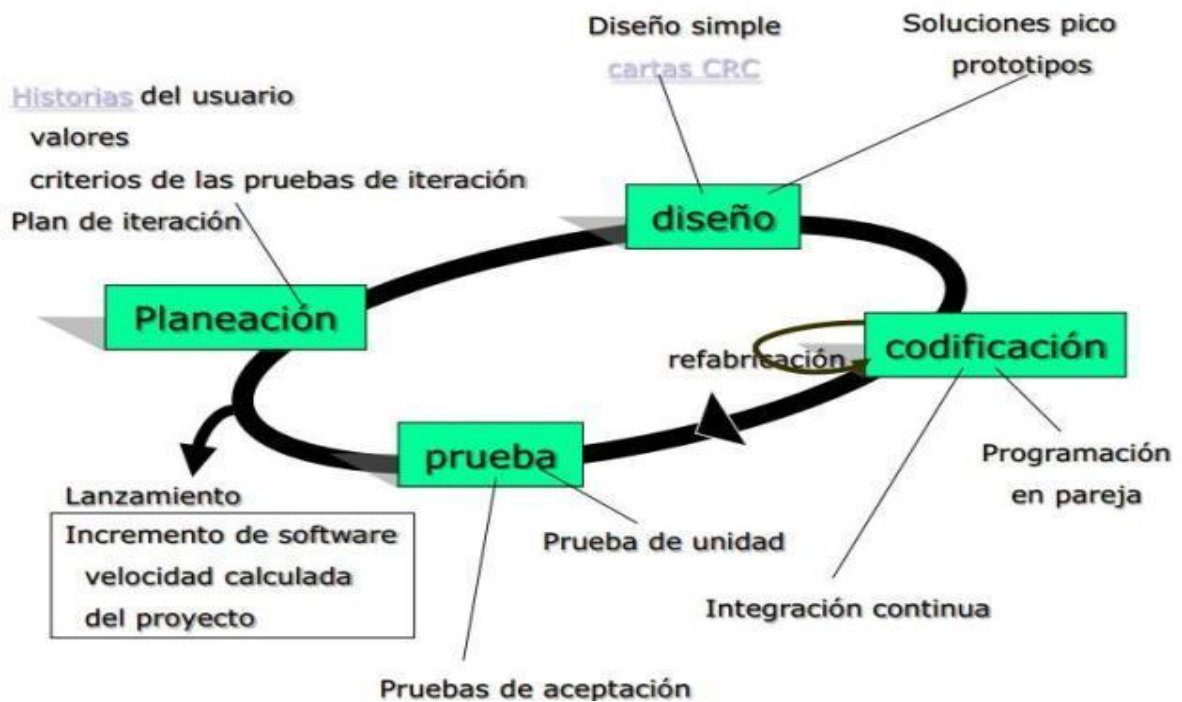


Figura 3. Fases de desarrollo metodológica XP

Fuente: Fases de desarrollo Metodología XP 2017 por Karla Cevallos

Las fases que la componen según Los autores Roux y Martín (2020).

- **Etapa 1 la planificación**

Sobre la base de la identificación del historial del usuario, las prioridades se establecen y subdividen en versiones más pequeñas. El plan se revisará aproximadamente cada dos semanas y se producirá un software funcional que se podrá probar y publicar.

- **Etapa 2 diseño**

Durante esta fase, intentaremos usar un código básico y haremos todo lo posible para reducir el trabajo de código. Obtendremos un prototipo y podremos agregar tarjetas CRC (clase-responsabilidad-cooperación) a nuestros diseños de software.

- **Etapa 3 codificación “de todos”**

Durante esta etapa, la programación se realiza en parejas, frente a la misma computadora; ocasionalmente, hay intercomunicación entre compañeros de trabajo; esto asegura que se genere un código más genérico, que cualquier otro compañero de trabajo pueda entender, dando como resultado un cronograma planificado.

- **Etapa 4 pruebas**

Debido a que a menudo son proyectos a corto plazo, las pruebas automatizadas y continuas son esenciales. Además, los clientes pueden probarse a sí mismos, sugerir nuevas pruebas y verificar la mini-versión.

- **Etapa 5 lanzamiento**

Si llegamos al final del proceso, significa que hemos probado con éxito todos los escenarios de usuario o versiones pequeñas, y los hemos ajustado a las especificaciones del cliente. También tenemos un software útil que se puede incluir en el producto.

Debido a que contienen varias fases dentro de su estructura, y este método se adapta y maneja de manera rápida y sencilla a la hora de documentar, la metodología XP puede resultar muy útil para su uso dentro del entorno de trabajo requerido. Además, tiene una característica muy sostenible de mantener la comunicación entre el desarrollador y los clientes, incluyendo la inyección de información valiosa en cualquier momento para realizar actualizaciones en las fases del proyecto, además de ceñirse a los valores de comunicación, sencillez y, por último, pero no menos importante, haciendo hincapié en la retroalimentación en cualquier momento.

2.2.12. Tipos de arquitectura de desarrollo de software

Según los autores Acevedo y Irrazábal (2020) existen varios tipos de arquitectura para el desarrollo de software, cada una de las cuales se enfoca en diferentes áreas o necesidades que el cliente está buscando para solucionar un problema que se está presentando dentro de su empresa o establecimiento. Echemos un vistazo a cuáles están disponibles.

- Cliente-servidor.
- Arquitectura por capas.
- Modelo vista controlador.

2.2.12.1. Arquitectura cliente servidor

“La arquitectura cliente-servidor se enfoca en mantener la comunicación entre el servidor y sus muchos clientes; esta arquitectura está orientada a proporcionar servicios organizacionales como bancos, unidades de crecimiento”. (Moyano, 2020)

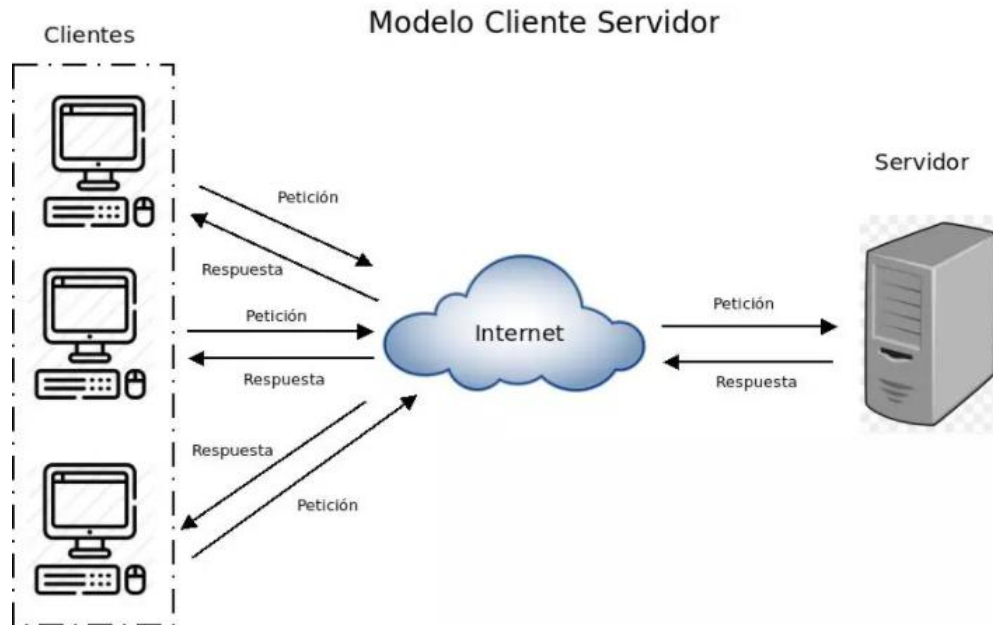


Figura 4. Función de la arquitectura
Fuente: Función de la arquitectura 2017 por Curí Human.

Teniendo en cuenta lo que mencionan los autores, la arquitectura cliente-servidor es un modelo de diseño de software donde las tareas se dividen entre el proveedor de recurso o servicio. El cliente envía una solicitud al servidor, y este último responde.

2.2.12.2. Arquitectura por capas

Según los autores Blanco y Rojas (2020) dicen que, la arquitectura de capas es exclusivamente para estructuras de subáreas que alimentan la mente superior y posterior, y está enfocada a la lógica de negocios, como la arquitectura de un banco. Si observamos la arquitectura de un banco, por ejemplo, usaríamos la separación de tres capas: "representación", "lógica empresarial" y "acceso".

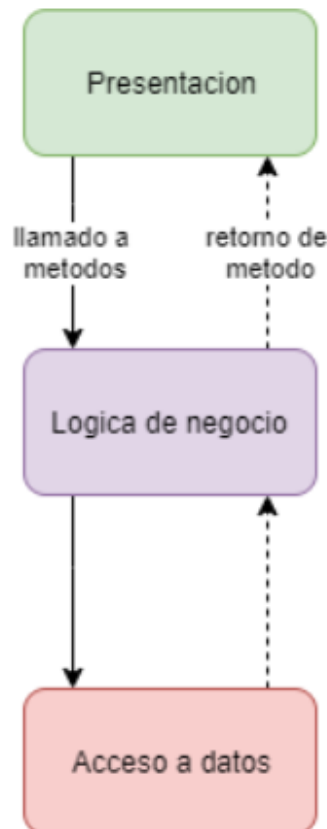


Figura 5. Función de la arquitectura

Fuente: Función de la arquitectura 2019 por Bertucelli Martin

Haciendo mención a lo que los autores definen. La arquitectura en capas incluye dividir la aplicación en secciones, con el objetivo de establecer un rol muy claro para cada una de ellas, como la de presentación, reglas comerciales y el acceso a datos. El estilo arquitectónico no define cuántas debe tener una aplicación, sino que se centra en separar la aplicación en varias a la vez.

2.2.12.3. Arquitectura Modelo-vista-controlador

Qué es el Modelo-Vista-Controlador

De acuerdo con el autor Carlos (2018), el MVC es una propuesta de arquitectura de software que se utiliza para separar el código en sus diversas responsabilidades, manteniendo distintas capas, cada una responsable de una tarea específica. Ha ganado mucha tracción y seguidores en los últimos años gracias a la aparición de numerosos marcos de desarrollo web que utilizan el patrón MVC, son las tres capas diferentes que componen el código, cada una con su propio conjunto de responsabilidades.

Modelo

Es la capa donde se trabaja con los datos, por lo que contará con mecanismos para acceder a la información, así como actualizar su estado. Los datos serán recopilados en una base de datos, por lo que tendremos funciones en los modelos para acceder a la tabla y realizar las siguientes acciones: seleccionar, actualizar e insertar datos. etc. (Orozco, 2019)

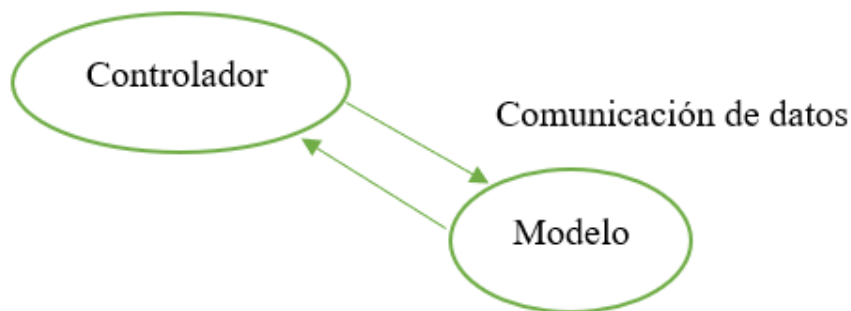


Figura 6. Función del modelo

Fuente: Funcionamiento del modelo 2019 por García Cobian

Vista

Las vistas, contienen el código de nuestra aplicación, este produce la visualización de las interfaces de usuario, o sea, el código que nos permitirá renderizar los estados de nuestra aplicación en HTML. En las vistas nada más tenemos los códigos HTML y PHP que nos permite mostrar la salida. (Castro, Herrera, y Villalobos, 2020)

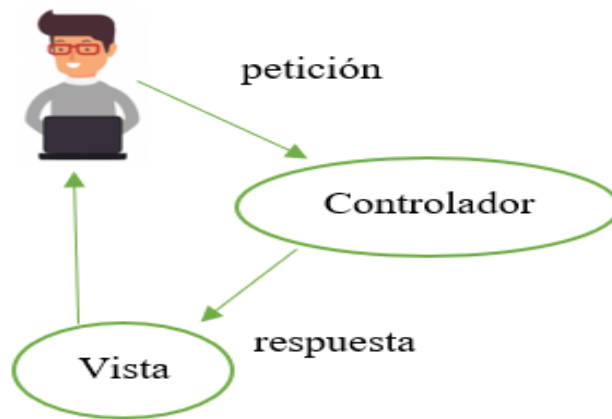


Figura 7. Función de la vista
Fuente: Funcionamiento de la vista 2018 por Eury Rodríguez

Controlador

Contiene el código necesario para actuar como intermediario entre el usuario y el sistema, respondiendo a acciones solicitadas por la aplicación, como ver un elemento, realizar una compra, realizar una búsqueda, etc. Realizar tareas de transformación de datos, como completar los campos de un formulario en línea utilizando el protocolo HTTP. (Rolando, 2017)

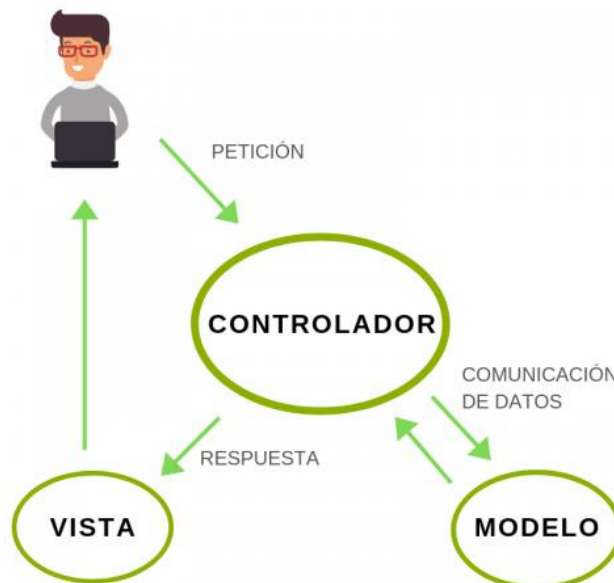


Figura 8. Función del controlador
Fuente: Funcionamiento del controlador 2019 por José María Aguilar

Cada una de las arquitecturas que se muestran son puestas en práctica según el área a la que vaya enfocada, como, por ejemplo, la arquitectura cliente servidor que está enfocada a la población y desarrollo de entidades bancarias, la siguiente arquitectura se divide en varias capas, con el objetivo de tener un rol muy claro para cada una de estas, donde tiene presentación comercial y por último el acceso a datos, el estilo arquitectónico no define cuántas capas debe tener una aplicación, al conocer sus propósitos estas ayudan a definir una solución para cumplir ciertos objetivos específicos y cumplir un general, donde recae todos los requisitos técnicos y operativos que se debe tener al momento de elegir y ser aplicada en el momento justo y necesario y llevarla a la implementación en su totalidad dentro de un proyecto, a continuación veamos la comparación de cada una de estas arquitecturas.

Tabla 3.

Comparación de arquitecturas de desarrollo de software

Arquitecturas de Desarrollo de software			
Características	MVC	C/S	NC
Escalabilidad	✓	✓	✓
Factibilidad de mantenimiento	✓	✓	✓
Flexibilidad	✓	✓	✓
Versatilidad	-	✓	✓
Control	✓	-	-
Solución	✓	-	-
Reusabilidad	✓	-	-
Seguridad	✓	-	-
Permite agregar nuevas funcionalidades	✓	✓	-

Nota. La tabla 3 muestra las características de cada arquitectura, véase el anexo 8.

La arquitectura MVC “Modelo Vista Controlador” es la que cumple con los parámetros para este proyecto, es la más indicada por su desempeño en el entorno web, combinada con nuestra metodología Xp que está orientada al entorno de desarrollo, por ende, esta comparación no discrimina a las demás arquitecturas por sus cualidades que no se halló en fuentes de información en los repositorios web de primera relevancia, dicho esto veamos la arquitectura del modelo vista controlador y sus especificaciones.

2.2.12.4. Arquitectura de aplicaciones MVC

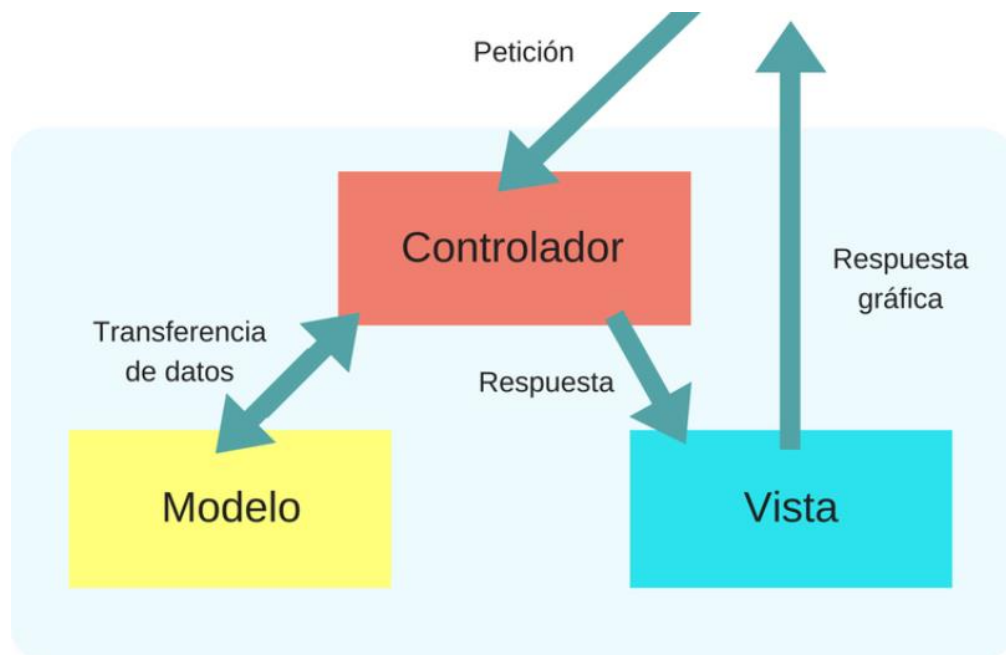


Figura 9. Arquitectura de Modelo Vista Controlador
Fuente: Arquitectura de Modelo Vista Controlador 2017 por García

Veamos lo que dice respecto a su arquitectura el autor García (2017).

- El usuario envía una solicitud en línea a nuestro sitio web. En general, cuando visite una página de nuestro sitio web, se le activará. La solicitud se entrega al supervisor.
- El controlador se comunica con el modelo y la vista. Solicite datos del modelo o envíe el modelo para actualizar los datos. Una vez concluida la operación respectiva, se solicitará a la audiencia correspondiente el cese de operaciones.
- Para generar una salida, la vista puede solicitar ocasionalmente más información del modelo, y el controlador será responsable de obtener todos los datos del modelo y enviarlos a la vista, actuando como conducto entre ellos.
- La vista invita al usuario a salir. Aunque, en algunos casos, la salida puede ser devuelta al controlador, que es quien inicia la entrega al cliente.

La arquitectura como se muestra en la parte superior nos indica como está establecido su funcionamiento y sus habilidades dentro de su arquitectura, todas interactúan entre sí para llegar a un solo objetivo que es cumplir con la petición del cliente.

2.2.12.5. Ventajas y desventajas de la arquitectura MVC

Rivera (2017), menciona las ventajas y desventajas de hacer uso de la metodología Modelo Vista Controlador.

Estas son sus ventajas

- La separación del modelo y la vista, o la separación de la representación visual de los datos y la vista.
- Es significativamente más fácil combinar muchas representaciones de los mismos datos o información.
- Ayuda en la adición de nuevos tipos de datos de acuerdo con los requisitos de la aplicación y es independiente de la operación de otras capas.
- Proporcionar un método más fácil de usar para probar el correcto funcionamiento del sistema.

Sus Desventajas son:

- La idea de las capas se suma a la complejidad del sistema.
- El número de documentos mantenidos y desarrollados ha aumentado significativamente.
- El nivel de aprendizaje y patrón de diseño es superior al de otros modelos básicos.

Sin embargo, se requieren varios componentes, como PHP, HTML y SQL, para que esta arquitectura funcione correctamente. Al usar MVC, puede usar bibliotecas como PDO, ORM y Doctrine. Esta arquitectura funciona con abstracciones de bases de datos, principalmente para evitar sentencias SQL. Algunos de ellos se basan en el motor de base de datos que está utilizando. El nivel de aprendizaje y patrón de diseño es superior al de otros modelos básicos, es la mejor opción que se adapta al desarrollo del proyecto que es la ya mencionada, se tiene que tener claro que no se escogió al azar, al preseleccionarla se debe tener en cuenta que esta arquitectura se debe manejar con su total compatibilidad con el lenguaje de programación y complementos que se vayan a utilizar como, framework, Base de datos, etc.

2.2.13. Lenguajes de programación

2.2.13.1. Lenguaje Java

Java es un lenguaje robusto, es el encargado de brindar interactividad y dinámica a las páginas web cuando se ejecuta en el navegador, no requiere un compilador, se lee el código directamente sin un tercero, este lenguaje es reconocido como uno de los tres lenguajes nativos de la web, es responsable de muchas funciones, como lenguaje de lado del cliente, orientado a objetos etc. (Falcon, 2018) también menciona las siguientes características:

- Java es un lenguaje codificado, compilado.
- Cuando se ejecuta java se verifica que sus clases estén escritas y programadas correctamente, es decir que no exista error de escritura de códigos y sintaxis.
- Java es uno de los lenguajes principales al venir ocupando los primeros lugares en el ranking sobre lenguajes de programación.

2.2.13.2. Lenguaje Python

Es un lenguaje de programación creado por Guido Van Rossum en 1989. Es conocido por su simplicidad y facilidad de aprendizaje, y varias instituciones educativas lo utilizan como base para enseñar a los estudiantes a programar. Debido a su versatilidad, se utiliza en una variedad de áreas de la tecnología de la información, incluida la web, la seguridad de la información, la ciencia de datos y la inteligencia artificial, entre otras, Python se ha convertido en una herramienta muy útil para desarrollar cualquier tipo de software a raíz de esto. (Talamé, 2019), Python es compatible con múltiples usuarios y tiene una gran cantidad de herramientas y módulos para el desarrollo de software. Entre estas características se encuentran las siguientes:

- Python es un lenguaje multiparadigma que le permite escribir código imperativo, funcional y orientado a objetos.
- En comparación con otros idiomas, su curva de aprendizaje es bastante baja.
- Debido a la simplicidad de su sintaxis, genera bloques de código concisos.
- Trabajar con estructuras de datos complejas es realmente simple con Python.

2.2.13.3. Lenguaje PhP

Php es un lenguaje de código abierto muy popular entre otros, especialmente adecuado para el desarrollo web, además de poder ser incrustado en HTML, el código de php está encerrado entre las etiquetas especiales de comienzo y final lo que distingue a php es su extrema simplicidad para el principiante, pero a su vez ofrece muchas características avanzadas para los programadores profesionales. (Mina, 2018), PhP es un lenguaje multipropósito que incluye una gran cantidad de herramientas y módulos para el desarrollo de software.

- El lenguaje PHP es uno de los más antiguos (fue creado en 1995).
- Se utiliza en la creación de páginas web que hacen uso de bases de datos.
- Es un lenguaje interpretado en el lado del servidor que permite la creación de páginas web dinámicas que pueden estar dentro de páginas en HTML.
- Es uno de los lenguajes más populares por su rapidez y la facilidad de desarrollo.

Por tanto, existe una inmensa cantidad de lenguajes hoy en día, muchos de ellos están enfocados en diferentes áreas, en este caso los lenguajes que se describen en la parte superior son orientados a desarrollo web, se ha visto que cada uno de estos se adapta a las necesidades que se están buscando dentro de un entorno de trabajo, para elegir uno de estos se debe conocer sus características, por esta razón veamos sus cualidades de cada lenguaje.

Tabla 4.
Comparación de los lenguajes de programación

Lenguajes de programación			
Características	Php	Python	Java
Gratuito	✓	✓	✓
Maneja su propio compilador.	✓	✓	Otros
Orientado a Objetos	✓	✓	✓
Multiplataforma	✓	✓	✓
Apoyo de aprendizaje	✓	✓	✓
Generar páginas web dinámicas	✓	✓	✓
Compatibilidad con MySQL	✓	✓	✓
Open Source	✓	✓	✓

Nota. La tabla 4 muestra las características de cada lenguaje de programación de desarrollo web, recopilación de varias fuentes, véase el anexo 9.

Al analizar el cuadro comparativo se establece la existencia de tres lenguajes de programación, donde tienen características muy parecidas entre ellas, pero no está dicho nada aun, por la razón que al elegir uno de estos de debe tomar en cuenta en qué entorno se lo va aplicar y en donde se enfocara en el desarrollo del proyecto donde intervienen similitudes de herramientas.

2.2.14. Framework de desarrollo web relacionados con PHP

2.2.14.1. Framework Laravel

Laravel es un framework de PHP de código abierto (Open Source) para el desarrollo de aplicaciones web, está familiarizado con Core PHP y Advanced PHP, además de ser de plataforma Mvc “Modelo-Vista-Controlador”, contiene la habilidad de instanciar clases y métodos para usarlos repetidas veces, este funciona con líneas de comando de Artisan que es el nombre de la interfaz donde ingresan los comandos para ejecutar funciones. (Ovando, 2019)



Figura 10. Logotipo de framework de laravel

Laravel es una herramienta con distintos fines donde se vea aplicada, es un framework de desarrollo web, define la estructura básica y además de su código que se utiliza para definir el contenido web, es capaz de brindar un código más simple y simplificado, La velocidad de carga de la página en el navegador, así como los elementos que contiene, son totalmente visibles en todos los navegadores.

2.2.14.2. Framework Slim

Slim es en realidad un micro framework para PHP y nos permite escribir aplicaciones y APIs de forma muy rápida y con muy poco código. No tiene la potencia de sus hermanos mayores (Laravel, Symfony, KumbiaPHP, ...) pero cumple bien su cometido. Y como buen framework, podemos vitaminarlo para aumentar sus funcionalidades, como dotarlo de un ORM con conexión a BDD, un gestor de templates como Twig, estos son sus aspectos, (Sunardi, 2019) también menciona sus características.

- Soporta métodos HTTP estándar y personalizados.
- Parámetros de ruta con comodines y condiciones.
- Renderizado de plantillas y vistas personalizadas.
- Encriptación segura de cookies con AES-256.

2.2.14.3. Framework Codeigniter V4

Es un framework de aplicaciones web de ámbito de código abierto, ayudando al desarrollo de programas en PHP, su objetivo es ayudar a los desarrolladores de proyectos de código, a desarrollar más rápido que escribir código desde cero, esto se logra a través de su amplio conjunto de bibliotecas para tareas comúnmente necesarias, teniendo su interfaz sencilla de acceso a las bibliotecas. (Muhamad, 2021), Ahora veamos las características a lo que menciona el autor.

- Sistema basado en Modelo-Vista-Controlador.
- Es de tipo ligero.
- Multiplataforma.
- Consta de marcos de plantilla.

Existe una inmensa cantidad de framework de estilo hoy en día, muchos de ellos están enfocados en diferentes aspectos, en este caso los framework que se describen en la parte superior son orientados con PHP, cada uno de estos se adapta a las necesidades que se están buscando dentro de un entorno de desarrollo, se debe tener claro en qué proyecto se está trabajando y que se quiere lograr al aplicar uno de ellos, para saber sus características veamos la siguiente tabla comparativa.

Tabla 5.
Comparación de framework de desarrollo web

Framework de Desarrollo Web			
Características	Code Igniter	Laravel	Slim
Basado en Modelo Vista Controlador	✓	✓	✗
Orientado a Objetos	✓	✓	✗
Manejo de plantillas	✓	✓	✓
Documentación completa	✓	✓	✓
Framework de aplicaciones web	✓	✓	✓
Manejo de MySQL	✓	✓	✓
Validación de datos de entrada	✓	✓	✗
Reglas de enrutamiento	✓	✓	✗
Fácil de lectura de código	✓	✓	✗

Nota. La tabla 5 muestra una comparación entre las características de los Framework de desarrollo web relacionados con PhP, recopilación de varias fuentes, véase el anexo 10.

Al analizar el cuadro comparativo se establece la existencia de tres Framework orientados al desarrollo web, teniendo características muy similares entre ellos, manejando varios entornos para el desarrollador, recordemos que estos son implementados según a lo que esté enfocado o a lo que se busca obtener con uno de estos. al utilizar estos frameworks se tiene ventajas al tener una amplia paleta de diseños únicos que se puede llegar a crear, haciéndolos ligeros y rápidos de interactuar con el usuario final. Al interactuar con varias fuentes en la búsqueda del framework Slim no se encuentra relación con algunas características, esto sucede por la falta de información en los repositorios web, pero esto no significa que no tenga estas cualidades dentro de su entorno.

2.2.15. Gestores de bases de datos

2.2.15.1. Gestor de base de datos PostgreSQL

PostgreSQL es un sistema de gestión de bases de datos para bases de datos relacionales de código abierto, aunque también se puede utilizar para consultas no relacionales. En este sistema, las consultas relacionales se realizan mediante SQL, mientras que las consultas no relacionales se realizan mediante JSON. Es un lenguaje de programación gratuito y de código abierto que admite tipos de datos avanzados y optimización del rendimiento. Estas características a menudo solo se ven en los sistemas comerciales de administración de bases de datos. (Vázquez, 2022), El uso de PostgreSQL ha crecido en popularidad en todo el mundo, y las razones principales de esto incluyen las siguientes.:

- Debido a que es de uso gratuito, lo utilizan desarrolladores de todo el mundo.
- Es un sistema multiplataforma con características únicas que lo diferencian de otros gestores.
- Atomicidad, consistencia, aislamiento y durabilidad son las características del soporte ACID.

2.2.15.2. Gestor de base de datos SQLite

SQLite es un gestor de base de datos relacional, contenida en una biblioteca muy pequeña, escrita en C, a diferencia de los otros gestores este no es un proceso independiente con el que el programa principal se comunica, sino la biblioteca SQLite se enlaza con el programa pasando a ser parte integral del mismo. (Deyimar, 2019)

El uso de SQLite se ha popularizado a nivel mundial, las principales razones por las cuales se ha incrementado son:

- Es de uso gratuito, implica que desarrolladores de todo el mundo lo utilicen.
- SQLite es una base de datos sin servidor y es independiente.

- Limitación de tamaño de aproximadamente 250 KB.
- SQLite almacena directamente información en un solo archivo.
- SQLite admite estos tipos de datos, BLOB, Integer, Null, Text, Real.

2.2.15.3. Gestor de base de datos MySQL

Es un sistema de gestión de base de datos relacional (RDBMS) de código abierto, basado en lenguaje de consulta estructurado (SQL), se ejecuta en prácticamente todas las plataformas, incluyendo Linux, UNIX y Windows. A pesar de que se puede utilizar en una amplia gama de aplicaciones, se asocia más con las aplicaciones basadas en la web y la publicación en línea y es un componente importante de una pila empresarial de código abierto llamado LAMP. es una plataforma de desarrollo web que utiliza Linux como sistema operativo, Apache como servidor web, MySQL como sistema de gestión de base de datos relacional y PHP como lenguaje de programación orientado a objetos. (Rouse, 2017)

Características

Veamos las principales cualidades que menciona el autor, Quinaluiza (2018)

- Le permite elegir entre una variedad de motores de almacenamiento para varias mesas.
- Un grupo de transacciones capaz de recolectar muchas conexiones para aumentar el número de transacciones por segundo.
- Una conexión segura.
- Ejecuta transacciones y utiliza contraseñas externas.
- Introducción a una parte importante del lenguaje SQL.
- Disponible en una amplia gama de plataformas y sistemas.
- Para la migración, utiliza una variedad de herramientas.
- Aclimatación temporal a la tabla hash.
- Al conectarse al servidor, proporciona una contraseña segura y un sistema de privilegios para la autenticación y comunicación del host mediante una contraseña cifrada.
- Se utilizan múltiples subprocesos a través del kernel.
- Acepta grandes cantidades de datos, incluso si el número total de registros supera los 50 millones.

Existe una inmensa cantidad de gestores de base de datos hoy en día, enfocados en diferentes aspectos como, flujo de datos de gran cantidad que ya no son simples bases de datos sino Datas, donde se analizan constantemente los datos que son almacenados en su entorno de trabajo, los gestores que se han descrito son enfocados a bases de datos relacionales, para saber que gestor

Se adapta mejor a la investigación tendremos que ver la siguiente tabla comparativa que se muestra a continuación.

Tabla 6.

Comparación de gestores de bases de datos

Características	SQLite	MySQL	PostgreSQL
Manejo de servidor de cliente	Incrustado	✓	✓
Servidor OS	✗	✓	✓
Escrito en C	✓	✓	✓
Multiplataforma	✓	✓	✓
DBaaS	✗	✓	✓
Multimodelo	✗	✓	✓
Relacional	✓	✓	✓

Nota. La tabla 6 muestra una comparación entre las características de los gestores de bases de datos. Recopilado por varias fuentes, véase el anexo 11.

Al analizar el cuadro comparativo se observa la existencia de tres gestores de bases de datos, teniendo características muy similares entre ellos, donde destacan MySQL y PostgreSQL en comparación de SQLite, dando así una idea clara de que podemos lograr con estos gestores, estos manejan varios entornos para el desarrollador, recordemos que estos son implementados según a lo que esté enfocado o a lo que se busca obtener con uno de estos.

2.2.16. Análisis de las herramientas y diseño de un sistema

El "análisis y diseño de sistemas" se refiere al proceso de examinar la situación de una empresa para gestionarla con métodos y procedimientos más adecuados, según los autores Carrizo y Alfaro (2018), a través de este contenido, un investigador podrá analizar y diseñar un sistema que proporcione la mejor perspectiva para la gestión del tiempo de la empresa a través de los controles, además un solucionador de problemas informáticos no solo debe saber cómo investigar lo que ya existe y cómo proponer un sistema racionalizado, sino que también necesita saber sobre los sistemas de información actuales, los principios de diseño, etc.

Según los autores Abrego, Tovar y Medina, (2017), "el análisis de un sistema de información es de gran importancia", y "el sistema de información produce un proceso confiable entre el analista y la computadora por la relación que existe entre ellos".

Según el autor Hidalgo y Camacho (2021) dicen que, la tecnología de la información ha jugado un papel fundamental en la automatización de procesos, con el objetivo principal de mejorar la eficiencia de la empresa en relación con las expectativas de los clientes. La computadora es la herramienta más importante en un proceso de automatización, y debe ser potenciada con la plataforma adecuada, siempre en línea con la visión corporativa y la estrategia de información. Con el paso del tiempo, la tecnología actual evoluciona. Como resultado, tanto las empresas grandes como las pequeñas utilizan herramientas para brindar un mejor servicio a sus clientes. Simultáneamente, debido a que la computación es un método de investigación, la ciencia de procesos se utiliza para el almacenamiento y procesamiento automático de datos e información. Obtener un diseño con varios beneficios que permitan tener una solución simple pero eficaz a la vez, al diseñar un software no se busca un diseño bonito que impacte a simple vista, sino que destaque por su funcionamiento, para esto existen varias herramientas que podrían ayudar en su correlación de lenguajes, pero no podemos mezclar varios lenguajes de distintas plataformas, sino herramientas que se encuentren relacionadas entre sí para tener un entorno más unificado.

III. METODOLOGÍA

3.1. ENFOQUE METODOLÓGICO

3.1.1. Enfoque cualitativo

En términos de metodología, el proyecto actual adopta un enfoque cualitativo. El uso de este estudio permitió la descripción de procesos relacionados con el control de inventario, ya que es un proceso inductivo contextualizado en un ambiente natural, esto se debe a que en la recolección de datos se establece una estrecha relación entre los participantes de la investigación sustrayendo información, a través de una entrevista no estructurada.

El enfoque es aplicable por la razón que contiene variedad de técnicas para levantar información ayudando así a tener información actual y real, también sometiendo a la empresa a un análisis que nos permita conocer los procesos y procedimientos actuales, mediante los siguientes métodos: recolección de datos, descripciones detalladas de situaciones, eventos, personas, interacciones, conductas observadas y preguntas textuales abiertas, en las que se analiza su necesidades y carencias, también donde el investigador se desplaza hacia el sitio de estudio para su recolección de información y perspectivas desde el punto de vista de quienes se involucran en la investigación.

3.1.2. Tipo de investigación

3.1.2.1 Investigación exploratoria

Las investigaciones de tipo exploratorias ofrecen un primer acercamiento al problema que se pretende estudiar y conocer, donde se establece una visión general sobre el tema incrementando la familiaridad con la temática estudiada al establecer las bases para una investigación más relevante con algo que hasta el momento desconocemos. (Nicomedes, 2017)

Esta investigación exploratoria es adaptable por ser flexible, está busca la identificación de la problemática que se está viviendo, donde implica que el investigador explore y establezca preguntas para responder en base del objeto estudiado, esta también se ve apoyada yendo de la mano con la investigación descriptiva.

3.1.2.2. Investigación descriptiva

La investigación descriptiva es la que se encarga de puntualizar las características de la población utiliza, tal como el nombre lo dice, para describir la realidad de situaciones, eventos, personas, grupos o comunidades que se estén abordando y que se pretenda analizar, el objetivo de esta investigación es describir el tema mas no descubrir el por qué ocurre. (Alberto, 2020) Esta investigación descriptiva es adaptable por aclarar y delimitar la problemática que se está viviendo en la investigación previa, apoyándose al determinar cuál es la situación a la que se va a estudiar con una breve descripción, también analiza el asunto o condición sobre el problema que se va a conocer al mismo instante que busca los detalles al examinar las características del problema.

3.2. IDEA A DEFENDER

Un sistema de inventario ayudará a mantener un control de los recursos que se utilizan en el proceso de compra y uso en la farmacia, del Hospital del Dia Nova Salud de la ciudad de Tulcán en el periodo 2021.

3.3. Operacionalización de variables

Tabla 7.

Operacionalización de la variable independiente " Sistema "

Variable	Definición	Dimensión	Indicador	Técnica	Instrumento	Informantes
Variable independiente Sistema	El sistema de control de inventario es una solución técnica que integra todos los aspectos de las tareas de inventario de la organización	Integración	- Número de módulos - Características del software.	Entrevista estructurada	Hoja técnica	Programador/ Administradora
		Casos de prueba	- Control de funcionalidades.	Evaluación	Prueba de aceptación	Programador
		Procesamiento	- Nivel de gestión de la configuración - Nivel de gestión de módulos - Nivel de aseguramiento de procesos	Matriz de evaluación	Matriz técnica	Programador

Tabla 8.*Operacionalización de la variable dependiente “Gestión del Inventario”*

Variable	Definición	Dimensión	Indicador	Técnica	Instrumento	Informantes
Variable dependiente Gestión del Inventario	Sobre el control de inventario se refiere a todos los procesos que conllevan al suministro, accesibilidad y almacenamiento de productos en alguna compañía	Manejo de procesos de la gestión de inventarios	<ul style="list-style-type: none"> - Inventarios. - Gastos sobre Compras. - Devolución sobre compras. 	Entrevista no estructurada	Guía de entrevista no estructurada	Administradora
		Fallas en la gestión de inventario	<ul style="list-style-type: none"> - Disponibilidad de materiales - Pronóstico de la demanda. - Niveles óptimos de inventario. - Rotación del inventario. 	Entrevista no estructurada	Guía de entrevista no estructurada	Bodeguero
				Observación simple no regulada	Ficha de observación	Bodeguero

3.4. MÉTODOS UTILIZADOS

3.4.1. Investigación cuasi experimental

El estudio cuasiexperimental es aquel en el que el sujeto de estudio no se elige al azar, sino que está predeterminado por el investigador. Se caracteriza por ser descriptiva, observando el comportamiento de los individuos e identificando también la forma en que se relacionan. (Álvarez, 2021)

Este método permitió pre seleccionar el grupo de estudio partiendo del origen donde ocurrían los hechos sin ser parte de su operación sino en ambiente neutral, donde se logró aplicar el instrumento que se denomina Hoja técnica donde se obtuvieron varios datos para ser analizados conjuntamente con los de la entrevista no estructurada.

3.4.2. Técnicas

3.4.2.1. Observación simple no regulada

El investigador se mantiene al margen del fenómeno estudiado, como un espectador pasivo, que se limita a registrar la información que aparece ante él, sin interacción, ni implicación alguna. Se evita la relación directa con el fenómeno, pretendiendo obtener la máxima objetividad y veracidad posible. (Guilln, 2017)

Esta técnica permitió el uso de la ficha de observación para demostrar varias características de su inventario, lo cual fue posible gracias al supervisor de bodega en el área de farmacia. Fue de suma utilidad para entender cómo se lleva a cabo cada uno de los procesos y qué parámetros se requieren; también permitió la categorización de los indicadores por color para determinar en qué nivel se encontraba cada indicador.

3.4.2.2. Entrevista no estructurada

La entrevista no estructurada es el instrumento que nos permitirá aproximarnos de forma más natural y abierta a los sujetos de estudio, sin que se sienta irrumpido ni examinado, de manera tal de ir logrando la confianza y el vínculo necesario que toda recolección de información necesariamente demanda. (Trindade, 2017)

Esta técnica permitió recoger datos importantes, aplicando la guía de entrevista no estructurada, obteniendo información sobre su control de su inventario que se lleva dentro de farmacia, arrojando datos favorables para la investigación donde se describió cómo se lo maneja actualmente.

3.4.2.3. Entrevista estructurada

“Una entrevista estructurada es aquella que se define como una herramienta para la investigación ya que es minuciosa en su ejecución y tiene poco o ningún espacio para alentar a los participantes a obtener y analizar resultados” (Escarcega, 2022).

Durante la investigación se le aplicó a la administradora una Hoja técnica, con el fin de obtener información sobre las características del software como, menús iconos, color etc. relacionados al diseño.

3.4.2.4. Matriz

La matriz es una herramienta técnica donde se puede evaluar parámetros estrictos que consiste básicamente en una subdivisión de los objetivos que se persiguen para una determinada tarea, de forma que cada componente pueda ser evaluado de forma explícita e individual el instrumento en sí puede estar en diferentes formatos. (Palacios, 2019)

Al aplicar esta herramienta con la técnica de matriz de evaluación se logró analizar el nivel del software que tiene actualmente para su funcionamiento, donde se lo evaluó por colores donde se indica que el color naranja es nivel medio al igual que el color amarillo se identifica como fuerte.

IV. RESULTADOS Y DISCUSIÓN

4.1. RESULTADOS

4.1.1. Análisis de los resultados obtenidos.

Este trabajo de investigación tuvo como finalidad sistematizar los procesos para gestionar la información, mediante el desarrollo de un sistema informático que permita ejecutar de mejor manera el control de inventario, conjuntamente con sus nuevas modalidades añadidas como, el módulo de Reportes, Financiero, Herramientas, Configuraciones etc.

En la aplicación de los instrumentos se logra evidenciar que existen varias falencias dentro de su área de trabajo que respecta al área de control y reabastecimiento de productos para su establecimiento.

En el levantamiento de requerimientos se aplicó una entrevista no estructurada a la señorita administradora y al encargado de bodega de la farmacia del Hospital del Dia Nova Salud de la ciudad de Tulcán. Donde se dio a conocer cómo se llevan las actividades de cada área relacionada con el control de su inventario, exponiendo sus procesos como, el de adquirir un nuevo producto, devolución y recibimiento de los Materiales y accesorios, a comparación de las demás actividades no se maneja una lógica para ser aplicados dentro de la organización, de igual manera al no tener una estructura clara y concisa para ser pasados al cuadro de programación, a estos se los puede denominar procesos genéricos.

Para el diseño de este sistema se integró herramientas de desarrollo web usando una arquitectura Modelo Vista Controlador, por lo cual, se utilizó el framework CodeIgniter4 con el lenguaje PHP y la base de datos MySQL, estas herramientas combinadas forman una sola, actualmente son usadas en grandes proyectos, los cuales cuentan con una gran comunidad que constantemente dan soporte y velan por su óptimo funcionamiento.

Se utilizó una metodología ágil para desarrollo de software llamada Xp, la cual permitió llevar un control de las tareas programadas, ayudando a clasificarlas en las tareas de usuario, haciendo que el investigador se enfoque en el área de más importancia que era el control de inventario de la farmacia del Hospital Del Dia Nova Salud. Mediante lo expuesto se logra evidenciar los resultados que arroja la investigación.

4.1.2. Entrevista dirigida a la Lic. Carmen Zúñiga administradora

Tabla 9.

Primera pregunta: Administradora

1. Pregunta	Entrevistado	Respuesta
¿A nivel general qué procesos se realizan en la clínica?	Administradora	<p>Se realiza el stock de la medicación para saber qué es lo que nos hace falta y según eso nosotros contactamos a los proveedores, se consulta precios y procedemos a la adquisición, en cuanto se procede a la adquisición del pedido, se saca y se verifica e ingresa directamente a lo que es farmacia.</p> <p>Procesos: adquisición del producto y verificación de stock.</p> <ol style="list-style-type: none">1. Se determina que medicación hace falta.2. Se contactan con los proveedores.3. Qué producto tiene el proveedor y que precio tiene.4. Se hace el pedido. <p>Proceso contable:</p> <p>Una vez que ya se recibe la medicación y se está de acuerdo con el pedido y con la cantidad y precios que nos llegan, la factura se le entrega a la señorita administradora.</p> <p>Obligaciones del SRI.</p> <p>El sistema contable es aparte del sistema de control de inventario.</p>

Análisis: El resultado obtenido muestra el proceso de adquisición de productos hacia los proveedores. Al momento de verificar los materiales en stock no se indican procedimientos claros como dicta la teoría para llevar un adecuado control de inventario. A continuación, mírese el flujograma para la adquisición.

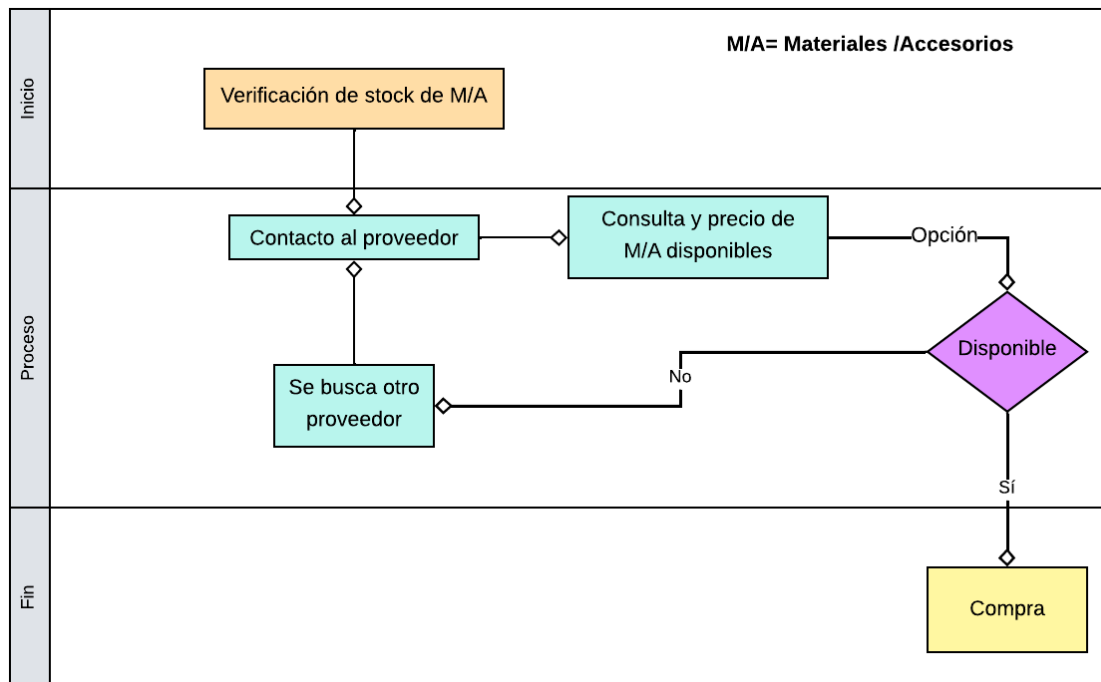


Figura 11. Proceso de adquisición de productos

Tabla 10.

Segunda pregunta: Administradora

2. Pregunta	Entrevistado	Respuesta
¿De qué forma se realizan los procesos administrativos sobre el control de inventario en el área de farmacia?	Administrador a	<p>Cuando se recibe la medicación, el encargado está en la obligación de recibir la mercadería y de contabilizar y verificar que nos llegue la cantidad correcta que fue solicitada. Los M/A se pasan directo a farmacia.</p> <p>Proceso de devolución</p> <p>A veces llega mercadería que no fue pedida y se procede a la devolución al proveedor.</p> <p>Se verifica de acuerdo al pedido que se solicita que llegue la medicación correcta y si no se está de acuerdo con el pedido se procede a devolver, se devuelve por fecha de caducidad, por diferente tipo de producto al solicitar inyectable y se nos envía en Tableta, esto se devuelve sin excepciones, las devoluciones se realizan de forma genérica, solo se contacta al proveedor y se procede a devolver el pedido, no se realiza ningún proceso interno para la devolución.</p> <ol style="list-style-type: none">1. Procesos de recibimiento de M/A2. Recibir los pedidos3. Contabilizar los M/A4. Verificar los M/A solicitados5. Se envía a farmacia6. Procesos de devolución7. Contacto al proveedor8. Devolución de M/A no solicitados

Análisis: La administradora manifiesta en cuanto a la devolución de pedidos solicitados por el proveedor que, se rigen a la factura que los proveedores entregan al igual que la hoja de pedido que emiten desde el Hospital, además, los M/A “Materiales y Accesorios” son enviados directamente a farmacia sin antes haberlos separado por categoría o clase, a continuación, mírese el flujograma de los procesos de devolución y recibimiento.

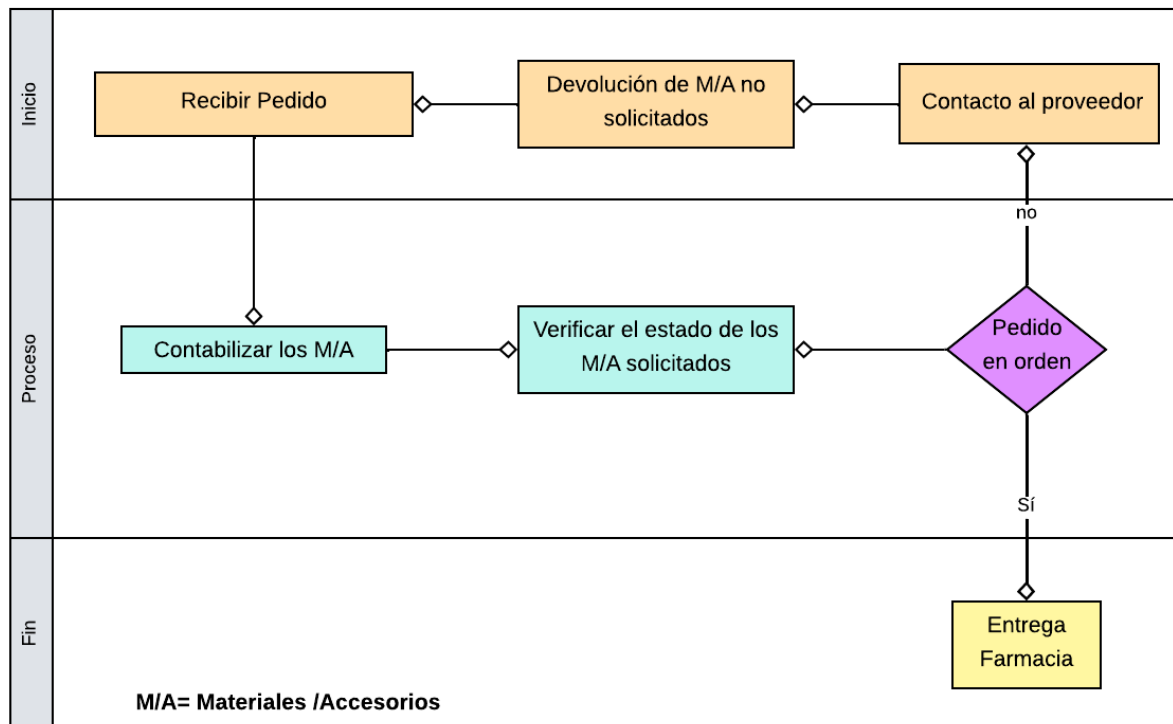


Figura 12. Proceso de devolución y recibimiento de los M/A

Tabla 11.

Tercera pregunta: Administradora

3. Pregunta	Entrevistado	Respuesta
¿Al no tener un control exacto de sus materiales y accesorios en bodega, como se lleva actualmente?	Administradora	<p style="text-align: center;">Control de bodega</p> <p>Antes se manejaba con un sistema por el cual se controlaba, pero generaba muchos problemas e inconvenientes, actualmente estamos con ese inconveniente que no tenemos un sistema y en realidad el control se lo hace casi manual, se verifica que es lo que se tiene y que no, entonces teniendo un control exacto no se lo tiene, más se lo realiza de una forma manual por ejemplo en Excel se lo maneja tipo Kardex, se dificulta llevar el registro en Excel, si nos interesa llevar un control, al igual que se lo capacite al compañero que trabaja ahí para que él pueda manejar el sistema, poder tener ya datos reales y evitar la fuga de medicación.</p>

Análisis: Se establece que la verificación es de forma manual, se realizan cálculos de manera tradicional, teniendo pérdida de tiempo y arriesgándose a tener un stock con datos irregulares, por ejemplo, llevarlo en un cuaderno o como bien lo dice la entrevistada en una hoja de Excel, poniéndoles en una posición no muy adecuada para saber con exactitud qué producto tiene más acogida que otro.

Tabla 12.*Cuarta pregunta: Administradora*

4. Pregunta	Entrevistado	Respuesta
¿Cuál es el problema de no contar con un registro exacto de los materiales y accesorios en el área de farmacia?	Administradora	El problema es que no se tiene un control rígido en cuanto a la medicación y no sabemos si existe o no la cantidad correcta, si existe fuga de medicación, se interesa tener un control exacto de la medicación mediante un sistema para tener valores reales, no se tiene un control exacto de la medicación en el Hospital.

Análisis: Esta respuesta indica que el departamento de farmacia no tiene un control adecuado en su totalidad, y como resultado, el departamento de farmacia se ve afectado, resultando en pérdida de medicamentos, demostrando que los procesos del departamento son ineficaces, dando como resultado datos erróneos en el Kardex que es generado. Con lo observado, es posible concluir que el departamento de farmacia está buscando ayuda.

Tabla 13.*Quinta pregunta: Administradora*

5. Pregunta	Entrevistado	Respuesta
¿De qué forma usted pensaría que un sistema ayudaría al área de farmacia a llevar un adecuado control de sus materiales y accesorios?	Administradora	Al implementar un sistema se podría revisar un x medicamento y ahí me va a dar exactamente un reporte de cuánto ingreso cuanto sale y cuanto debo de tener, porque a veces el tiempo no alcanza y se complica por cualquier cosa y ya no se registra, el sistema muestra datos reales y si el sistema me dice que tiene 10 aspirinas deben estar las 10 aspirinas, entonces sé que está bien, pero al momento de llevarlo manual no se logra tener un control real y como sabe el papel aguanta todo y siempre hay esa desconfianza que si se está manejando bien o mal la farmacia porque no hay un control rígido.

Análisis: Al apoyarse en una herramienta tecnológica se podrían obtener datos reales para su debido uso como puede ser, el registro de salida y entrada de los materiales y accesorios, ayudando a generar reportes de ventas, compras y por último un reporte general del inventario, dejando a un lado el registro convencional para llevarlo a un nuevo nivel dentro de un sistema organizado.

Tabla 14.*Sexta pregunta: Administradora*

6. Pregunta	Entrevistado	Respuesta
¿De qué manera se realiza el detalle de los precios de venta al público?	Administradora	Nosotros prácticamente tenemos muy poca venta al público, lo que nosotros tenemos es más para la administración de los pacientes, entonces prácticamente del precio que nos viene a nosotros en la factura nosotros le incrementamos un porcentaje del 12 % para poder nosotros vender los medicamentos y se los administra a los pacientes para caso de hospitalizaciones.

Análisis: En esta pregunta se indica cómo se maneja el detalle o incremento del IVA, se incluye el 12% directo a los precios que vienen marcados en las respectivas facturas, luego de haberse sumado el porcentaje pasa a la venta y también a la hospitalización de los debidos pacientes del Hospital, esto se puede pasar al cuadro de programación al haberse mencionado dentro de esta pregunta.

Tabla 15.*Séptima pregunta: Administradora*

7. Pregunta	Entrevistado	Respuesta
¿Cómo se realiza la compra de recetas de pacientes internos?	Administradora	Para los pacientes internos no se hace receta, nosotros nos guiamos en las indicaciones médicas que dispone cada médico tratante allí, el médico deja especificando, por ejemplo, “la ampicilina una cada 8 horas entonces más o menos hacemos cuentas los días que esté el paciente, las penicilinas tienen que estar colocadas y así se hace con todos los medicamentos con todo lo que el médico deja estipulando que se le coloque al paciente eso queda inscrito en la historia clínica “digamos que arriba en hospitalización se utilizó 10 ampicilinas, entonces al momento de que se hace la cuenta para cobrar al paciente , esas 10 ampicilinas repone farmacia a enfermería porque en enfermería también se maneja un inventario pequeño de medicación, es uso estricto de los pacientes que están en hospitalización

Análisis: se pone en conocimiento la existencia de un segundo departamento, enfermería, dependiente de farmacia, se lleva un control de inventario pequeño y se distribuye medicamentos a esa área que salen exclusivamente de farmacia hacia enfermería, siendo así esta quien abastece a este departamento la medicación, además se aclara que no se realizan recetas para los pacientes internos.

Tabla 16.*Octava pregunta: Administradora*

8. Preguntas	Entrevistado	Respuesta
¿Cómo se determina la fecha de caducidad de los materiales y accesorios?	Administradora	Prácticamente en todos los productos nos vienen con fecha de caducidad, entonces nosotros tenemos la semaforización que obliga el ministerio de salud para la medicación, según eso se semaforiza, los de color verde tienen mucho tiempo de caducidad, amarillo ya están con poco tiempo, y los que están con rojo están próximos a caducarse, no se tiene un inventario total como se tendría en una farmacia local si no acorde de la hospitalización de los pacientes.

Análisis: Teniendo presente la respuesta que se dio a conocer sobre el registro de las fechas de caducidad de los M/A, se muestra una técnica por el ministerio de salud para el registro de caducidad de estos medicamentos donde se dividen en tres colores, el primero es el verde, este tiene una estimación de un año o más, el amarillo, tiene promedio de 6 a 11 meses, y el último es el rojo, tiene una estimación de menos de 6 meses. También se nos muestra que no se tiene un riguroso inventario como se lo vería en cualquier farmacia de la ciudad, sino que es acorde a la necesidad de la hospitalización de los pacientes.

Tabla 17.
Novena pregunta: Administradora

9. Pregunta	Entrevistado	Respuesta
¿Qué tipo de inventario se utiliza en el área de farmacia? Para su mayor comprensión véase la siguiente información:	Administradora	Se utiliza en rotación del inventario, procedimientos internos más para la farmacia de uso interno de los pacientes que están hospitalizados, se utiliza el tipo B. en enfermería y farmacia, la venta al público es muy poco, la farmacia en sí es más para los pacientes que están hospitalizados, solo existen dos departamentos que son farmacia y enfermería.

A	B	C
Productos de alto valor, de gran venta, que requieren de mayor atención y cuidado a través de:	Productos de alto valor con ventas moderadas, requiere un tratamiento normal; es decir una atención ajustada a los requerimientos del negocio	Productos de bajo valor y poca venta, que deben tratarse según el principio de la simplificación productiva y administrativa y de reducción de costos.
Análisis de mercado, de precios y costos		Requerimientos simplificados de inventarios
Registro y control de inventarios		Trámites simplificados en el manejo de pedidos y pedidos de grandes cantidades
Determinación precisa de seguridad		Supervisión de las existencias
Aplicación preferencial del análisis de valores		

Análisis: Se menciona que se utiliza la categoría tipo B, este se maneja de una forma normal donde se adapta a las necesidades del negocio, este es aplicado en los dos departamentos para uso estricto de los pacientes internados, teniendo en cuenta que los inventarios tanto de farmacia como de enfermería no son robustos a comparación de las demás farmacias locales de la ciudad.

4.1.3. Entrevista dirigida al señor Enriques Telpiz “Bodeguero”.

Al aplicar la entrevista se mencionó que el señor Enriques Telpiz es el encargado de dos departamentos que son, farmacia y bodega, donde recibe los materiales y accesorios para su debida revisión al llegar al establecimiento, luego procede a la colocación de sus debidas perchas, véase a continuación los datos recolectados.

Tabla 18.

Primera pregunta: Bodeguero

1. Pregunta	Entrevistado	Respuesta
¿Tiene acceso a internet la bodega y las áreas respectivas como despacho y recibimiento de productos?	Bodeguero	<ul style="list-style-type: none"> - Se tiene acceso a internet en las áreas que se nombraron. - Cuando llega el producto hay que verificar que llegue completo, hay situaciones que despachan mal, se hace la verificación y se va ingresando al sistema. Por el momento se verifica que esté completo y el valor y las fechas de caducidad. - Hay productos que se piden por cajas en medicina, ejemplo, ampicilina. - Pedidos por unidades, ejemplo, bicarbonato de sodio. - Otra es de insumos “guantes jeringas catetes sondas algodones” - Se tiene por numeración para las perchas una para medicamentos y en la otra va los insumos. Pequeña percha de psicotrópicos, estupefacientes - Número de perchas - Percha de estupefacientes y psicotrópicos - Segunda insumos médicos - Por fechas de medicación.

Análisis: Se indica que una sola persona maneja estas dos áreas, es el encargado de revisar y de colocar cada material y accesorio en su debida percha, se muestra un proceso que lo realizan al momento de llegar medicación, este método no es el adecuado para manejar un registro real de la nueva mercadería, como se evidencia son procesos que conllevan a generar datos erróneos al momento de llevar un control estricto dentro del departamento.

Tabla 19.
Segunda pregunta: Bodeguero

2. Pregunta	Entrevistado	Respuesta
¿Usted tiene conocimiento y dominio de los tipos de inventario que existen como son los tres tipos ABC? Véase la siguiente información	Bodeguero	El entrevistado menciona que se maneja el tipo de inventario de la categoría B. donde muestra conocimiento por sus años de experiencia.

A	B	C
Productos de alto valor, de gran venta, que requieren de mayor atención y cuidado a través de:	Productos de alto valor con ventas moderadas, requiere un tratamiento normal; es decir una atención ajustada a los requerimientos del negocio	Productos de bajo valor y poca venta, que deben tratarse según el principio de la simplificación productiva y administrativa y de reducción de costos.
Análisis de mercado, de precios y de costos		Requerimientos simplificados de inventarios
Registro y control de inventarios		Trámites simplificados en el manejo de pedidos y pedidos de grandes cantidades
Determinación precisa de seguridad		Supervisión de las existencias
Aplicación preferencial del análisis de valores		

Análisis: A través de esta pregunta se indica que se trabaja con la categoría B, esta recae en su inventario al ajustarse a las necesidades que tenga el establecimiento de salud de acuerdo a sus pacientes hospitalizados, la rotación de inventario es interna a comparación de la venta al público. Esto ayuda a tener una mejor idea de cómo se lo puede implementar dentro de un software, teniendo un inventario más sencillo de manejar.

Tabla 20.*Tercera pregunta: Bodeguero*

3. Pregunta	Entrevistado	Respuesta
¿De qué forma usted tiene conocimiento y dominio de un computador para hacer uso de un nuevo software en una escala del 1 a 10%?	Bodeguero	-Prácticamente ahorita no manejamos un sistema, lo que manejamos es en Excel, no se cuenta con un sistema específico para este manejo de bodega. -Se tiene dominio de un computador un nivel básico que sería un 6%

Análisis: El resultado expuesto en una escala del 1 al 10% muestra un porcentaje de aceptación del 6%, con esto se deduce que está dentro del rango establecido para el manejo de un nuevo sistema, caso contrario se vería la necesidad de aplicar más horas para el buen uso del software.

Tabla 21.*Cuarta pregunta: Bodeguero*

4. Pregunta	Entrevistado	Respuesta
¿Cuáles son los pasos que usted realiza al momento de recibir un pedido?	Bodeguero	- Primeramente, cuando llega las cajas - Contabilizar los productos - Firmar el documento de entrega de guía de emisión - Almacenar (cajas grandes) - Sacar la factura y verificar que todo el producto esté completo y esté en buen estado. - Se va ingresando en Excel, Cantidad, Cambio de precio - Perchando

Análisis: En esta pregunta se indican los pasos que se tiene para recibir el producto y empezarlos a colocar en sus debidas perchas, si bien ya se mencionó en la parte superior uno de ellos, se vuelve a indicar otro, aclarando que no se tiene procedimientos fijos para regirse en su totalidad, se puede llegar a mejorar y regir un solo para todos los departamentos, llevando a un mejor desempeño en todas sus áreas que sea requerido.

Tabla 22.*Quinta pregunta: Bodeguero*

5. Pregunta	Entrevistado	Respuesta
¿Cómo es el manejo del control de existencia en los materiales y accesorios que se encuentran en farmacia?	Bodeguero	<ul style="list-style-type: none"> - Cuando ya tenemos una cierta cantidad solo se le suma siempre y cuando sea del mismo distribuidor, se coloca el nombre del distribuidor. - Si es nuevo producto y nuevo distribuidor - Nombre del producto - El precio - El están

Análisis: A través de esta pregunta se indica el proceso de control de existencias dentro de farmacia, se le suma al valor ya presentado anteriormente a la nueva cantidad siempre y cuando sea del mismo proveedor, caso contrario se agrega un nuevo registro con el nombre del nuevo medicamento.

Tabla 23.*Sexta pregunta: Bodeguero*

6. Pregunta	Entrevistado	Respuesta
¿Cómo se controla la entrada o salida de los materiales y accesorios adquiridos?	Bodeguero	<ul style="list-style-type: none"> - Por medio de la historia clínica o por medio de una hoja de tipo Kardex de cada descargo de historia clínica cuando es venta. - El paciente viene a la consulta y el médico le administra el tique que se relocaliza un pedido de enfermería y se paga en farmacia. - Maneja las dos categorías. - Maneja los dos stocks

Análisis: En esta pregunta se menciona que la persona delegada de farmacia maneja el inventario de enfermería, después de hacer uso de la medicación de enfermería se tiene que subir hacer recuento de los materiales y accesorios que fueron utilizados y abastecer este pequeño inventario, si ponemos esto en una balanza es una pérdida de tiempo subir y hacer el conteo manual de los productos que fueron administrados por las señoritas enfermeras.

Tabla 24.

Séptima pregunta: Bodeguero

7. Pregunta	Entrevistado	Respuesta
¿Qué método se utiliza Fifo/Lifo para los materiales y accesorios?	Bodeguero	Se utiliza el método Fifo para la salida de los materiales y accesorios

Análisis: Se logró comprender el método que se utiliza para la venta de los materiales y accesorios, se indicó que se utiliza el método Fifo, recordemos que este método se utiliza cuando un material o producto llega y sale enseguida a esto se lo conoce como “el primero en entrar, primero en salir” entonces concuerdan con procesos que se nos fue indicados en el transcurso de la entrevista, por ejemplo, en el tipo de inventario de categoría B se combina con este método y hacen hacer un buen uso para su inventario si lo llevaran de una manera correcta conjuntamente con sus procedimientos.

4.1.4. Matriz técnica del nivel de soporte del sistema

Tabla 25.

Matriz de Sp 1.1

Área de proceso	Soporte promedio	N de entradas soportadas	Soportado por BD	Nivel de soporte
1. Gestión de la configuración	5	10	100	Medio
2. Gestión de ventas	6	15	255	Fuerte
3. Gestión de Compras	5	15	333	Fuerte
4. Gestión de catálogos	6	10	255	Fuerte
5. Gestión de herramientas	6	11	255	Medio
6. Gestión de finanzas	4	13	150	Medio
7. Gestión de inventario	9	25	200	Fuerte
8. Gestión de reportes	7	16	100	Fuerte
9. Gestión de administración	5	13	100	Medio
10. Gestión de requerimientos	2	8	100	Medio
11. Aseguramiento de procesos válidos	110	84	1010	Fuerte
12. Medición %	8.2	8.0	37.8	Fuerte
13. Módulos sin funcionamiento	Ninguno	Ninguno	Ninguno	Fuerte
14. Valorización en color				
15. Nivel de riesgo	1.1	1.3	4.2	

16. Valoración total	174.3	229.3	2.600
17. Menos nivel de error	172.38	226.32	2.490,8

Tabla 26.
Matriz de resultado Sp 1.2

Área de proceso	Soporte promedio	Tipo de soporte
1. Gestión de la configuración	6.28	Fuerte
2. Gestión de módulos	7.8	Fuerte
3. Aseguramiento de procesos	8.33	Fuerte

Tabla 27.
Matriz de respuesta por porcentajes

Sp	Prof1	Prof2	Prof3	Prom
1. Sp1.1-2	87,5 %	75.6%	63.6%	95,8%
2. SBD2.1	86.6%	86%	70%	75%

Tabla 28.
Matriz de resultado final de las áreas de proceso

Área de proceso	Soporte (resultado de evaluación)
1. Gestión de la configuración	Si, para las dos prácticas específicas consultadas
2. Gestión de módulos	Si, para las dos prácticas específicas consultadas.
3. Aseguramiento de procesos	Si, para sp 1.1-2 Si, para sp 2.1

El resultado que arrojó la matriz es favorable para hacer uso del nuevo sistema y sus componentes, se evaluaron funciones del software, gestión de configuraciones, módulos y procesos. Dando como resultado porcentajes favorables indicadores de un sistema apto para su funcionamiento, para esta evaluación se utilizó como referencia la entidad w3c, este genera una matriz en los medios que se requiera evaluar.

Esta entidad ayudo a medir sobre porcentajes y colores su estabilidad de uso, especificándose en colores como el tomate que indica que se tiene un nivel medio, por otro lado, el color amarillo indica que es fuerte el nivel evaluado, de esta forma se evaluó las distintas áreas del sistema, el nivel promedio del software es fuerte relativamente bajo la tasa de error que es el 1.1 que limitaría sus funciones de uso. Los porcentajes arrojados son favorables donde se tiene el casi 100% de rigidez de estabilidad del software que se lo fabricó bajo una plantilla adaptando módulos y funciones, estos resultados demuestran que el software está listo para usarse dentro de una organización.

4.1.5. Hoja Técnica

Tabla 29.

Hoja técnica

Hoja Técnica		
Baribales	Características	Indicador
1. Menús e iconos	Propósitos de los menús y los iconos (reducen el tiempo/energía para comprender u navegar en el software) Apariencia de los menús y los iconos (son de rápida significación e interpretación)	Acceso a los menús e iconos (las instrucciones son claras para su selección) Los iconos son de tamaño y color adecuados, reconocibles y legibles
2. Diseño de la pantalla	Criterio de despliegue de la pantalla. Hay un título en cada pantalla. Se tiene en cuenta el movimiento natural del ojo.	Criterio de interacción de la pantalla con el usuario Las áreas de datos son obvias. Se usan pausas cortas.
3. Color	Se ajustan al descanso visual del ojo, por ejemplo: azul, blanco, rojo, amarillo, verde, tomate, plomo	Se usan más los colores legibles del centro del espectro (tomate, azul, verde, rojo, plomo)
4. Gráficos	Proporcionan suficiente detalle para cubrir la información necesaria	Se adaptan al nivel de conocimiento
5. Animaciones	Forma regular	Aumentan el aprendizaje sin distracciones
6. Textos	Estilo y legibilidad. Comprensión del texto	El tamaño de una línea está entre 1 y 5 palabras La lectura inicial del material es fácil de leer
7. General	Se adapta a los dos módulos solicitados	Control de inventario y abastecimiento

8. Contenido	El contenido es preciso	Actual y adecuado a las especificaciones
9. Adecuación	La aplicación es adecuada para su uso	Se adapta módulos no solicitadas
10. Técnico	El software se ejecuta sin retraso indebidos.	Las interfaces son limpias. El gráfico expuesto es claro y preciso
11. Claridad	La interfaz es sencilla y de fácil uso.	La secuencia de los elementos del menú es lógica. El usuario requiere un mínimo de conocimiento informático para trabajar con el programa.


Gracias a esta matriz se pudo obtener información en aspectos técnicos para el desarrollo del software obteniendo características como uso de colores adecuados y colocación de menús claros y sencillos que sean fáciles de utilizar, con estos aspectos se logró ubicar todo lo que fue informado en la hoja técnica dentro del nuevo sistema.

4.1.6. Ficha de observación

Tabla 30.

Ficha de observación del manejo de inventario

Ficha de Observación				
Manejo de inventario				
ITEM	INDICADOR	NIVEL DE ALARMA		
		VERDE	AMARILLO	ROJO
		Siempre	Algunas veces	Nunca
1	¿El bodeguero verifica si un elemento que ingresa al almacén tiene un código establecido?			X
2	¿El bodeguero instala los productos en un sitio adecuado?		X	
3	¿El bodeguero lleva un registro del ingreso de dichos materiales?		X	
4	¿El bodeguero lleva un registro de los productos solicitados?	X		
5	¿El bodeguero lleva registro de todas las salidas de los productos del almacén?	X		
6	¿El bodeguero recibe los vales u orden de consumo?		X	
7	¿El bodeguero lleva el registro que relaciona los materiales entregados al otro departamento de enfermería?		X	

8	¿Diariamente el bodeguero consolida todas las órdenes de consumo, salidas y las entradas enviando la información a la administradora?		x		
9	¿El bodeguero, verifica mensualmente si los materiales relacionados en las entradas y salidas, se encuentran en el almacén mensualmente, mediante inventario físico?		x		
10	¿En la oficina central, se reciben requisiciones de pedidos?		x		
11	¿Se llevan Kardex de elementos de alta rotación?			x	
12	¿El bodeguero verifica el producto y la cantidad del mismo, e informan la conformidad?			x	
13	¿En la oficina central se ha podido consolidar un registro diario de todos los materiales que se encuentran en stock?			x	
OBSERVACIONES		Total, Preguntas	2	7	4
Los procesos que se llevan actualmente no son los adecuados		Puntos por pregunta	1	2	3
		Resultado			28
VERDE: Se cumple			>=35		
AMARILLO: Acciones preventivas			<=35		
ROJO: No cumple, hay que tomar acciones correctivas			<=35		

Todo este proceso arrojó resultados valiosos para realizar el diseño de la aplicación, llevando los datos obtenidos a un modelo de ejecución de procesos diseñados a medida en el cual se enfocó el control de inventario, por otro lado el presente proyecto se basó en un enfoque de desarrollo de modelo vista controlador, llevándolo a otro nivel de desarrollo que es un sitio web, desarrollado con PHP codeigniter versión 4, donde se utilizó MYSQL como motor de base de datos y como gestor PhpMyAdmin, combinando tecnologías Frontend como Bootstrap, HTML5, CSS y JavaScript para mejorar la experiencia del usuario y ampliar la cantidad de personas que pueden acceder a la información.

Es necesario considerar la conectividad, con los sistemas de escritorio, el acceso a la información es sencillo porque el software se ejecuta localmente en las computadoras; por otro lado, una aplicación web requiere un servidor para ejecutar los servicios que conectan a los clientes a través de Internet.

El objetivo de un sistema de control de inventario es centralizar los datos, por lo que se prefieren las plataformas web porque son más fáciles de escalar e integrar con varios dispositivos. Este enfoque se traslada al proyecto actual y se traduce en los procesos que el sistema integra dentro

del Servicio para la Farmacia del Hospital de la Nova Salud, adaptando los módulos necesarios al sistema, en respuesta a la avalancha de datos que generan estos dispositivos, se dice que el sistema de control de inventario, en conjunto con sus funciones, está orientado a hacer un sistema escalable, además de adaptarse a una variedad de necesidades.

4.2. DISCUSIÓN

El objetivo de la investigación, que es desarrollar un sistema de gestión de inventarios para llevar un control adecuado de la farmacia del Hospital del Dia Nova Salud de Tulcán, sirvió de base para la discusión.

Se revisó antecedentes de investigaciones pasadas para analizar sus problemáticas, circunstancias, herramientas aplicadas etc., donde se comparte ciertas características con la investigación que se está abordando, una de ellas es fijar un control estricto dentro del área que se vea afectada, buscando ayuda en herramientas tecnológicas, este fue el foco principal de los tres antecedentes que se los tomó como referencia para basarse en una solución técnica.

El uso de un enfoque de investigación cualitativa permitió una entrevista no estructurada con la administradora, comenzando con la recopilación de datos para crear un marco teórico y metodológico que servirá como guía para el desarrollo de una plataforma tecnológica que administrará los materiales y accesorios del departamento de farmacia, como el control de inventario, donde se lleva a cabo procesos de registro, venta y abastecimiento, estos son llevados a cabo en notas de papel y hojas de Excel dificultando su operacionalidad, con la actual aplicación que se desarrolló se podrá dejar a un lado estas herramientas coloquiales llevándolos a un nuevo nivel de registro donde se los podrá llevar de una mejor manera y controlada, donde sus funciones ayudaran a tener un mayor flujo de información actualizada y obteniendo datos reales al ser accesible en cualquier momento.

También se aplicó una entrevista no estructurada basada en una guía de preguntas al bodeguero del hospital, con su ejecución se identificó los principales parámetros que se debe incorporar en el sistema, por otro lado, se analizó los procesos administrativos que se llevan actualmente con la finalidad de separar los procesos específicos que irían en el sistema, como también se analizó los procesos que ayudarían a tener un mayor rendimiento, estos son, el módulo de catálogos, Herramientas, Finanzas, Reportes y Administración, recordemos que estos módulos no fueron solicitados sino que fueron adaptados de una forma básica en el sistema.

La adaptación de estos segundos módulos serviría para ser escalable al sistema en futuras actualizaciones. De igual forma se obtuvo varios datos del tipo de inventario que se lleva en la farmacia a la hora de vender y a la hora de reabastecimiento este mismo, donde se utiliza la categoría B “Productos de alto valor con ventas moderadas, requiere un tratamiento normal; es

decir una atención ajustada a los requerimientos del negocio”, estos datos fueron de gran utilidad en el cumplimiento del objetivo del proyecto. Con los resultados presentados, se ha establecido un punto de referencia para trabajos futuros que pueden ser utilizados como base para el desarrollo de software para evaluar el impacto potencial de su implementación en el área de estudio u otros departamentos que están relacionados con los procesos de control de inventario en futuras investigaciones.

V. CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES

- La información recopilada a través de la base teórica permitió Identificar los procesos con falencias que tiene el Hospital del Día Nova Salud para el control del inventario, estos son, proceso de adquisición de productos, proceso de devolución y recibimiento de los materiales y accesorios, verificación de forma manual y por último los cálculos se los realiza manualmente, de esta forma se evidencia que no se tienen una forma adecuada para realizar un adecuado control de su inventario.
- Los procesos administrativos que se hallaron con falencias fueron analizados conjuntamente con sus requerimientos la cual se adaptan fácilmente al sistema por estar relacionados con las distintas áreas de la organización, la conveniencia de buscar y procesar se puede optimizar de forma centralizada utilizando sus nuevas herramientas, traduciéndolas en los módulos y complementos del sistema.
- Las herramientas de XP son muy útiles en el desarrollo de sistemas de control de inventario, al permitir descomponer los módulos de software según su importancia, además dar fecha límite para asegurar suficiente organización para los proyectos, a la vez prioriza la entrega de las actividades más relevantes. La utilización de software libre en el desarrollo del sistema que se enfoca en un sitio web con modelo de desarrollo Modelo-Vista-controlador, dio los resultados esperados, la combinación de funciones que integra y centraliza conjuntamente con los componentes de un sistema sólido que proporciona Información utilizable.

5.2. RECOMENDACIONES

- La investigación se centró en la creación del prototipo; por lo que se recomienda ampliar el proceso investigativo, tomando como referencia la documentación generada en este proyecto, y viendo su posible implementación conjuntamente con otras áreas.
- Estudiar los procesos administrativos de las áreas adyacentes al departamento de Farmacia y analizar la posible incorporación de más módulos, es recomendable realizar este proceso de forma progresiva para evitar fuertes impactos de acoplamiento entre los componentes del software.
- Dentro de la metodología XP es recomendable mantener una comunicación constante con el cliente para comprobar que los requerimientos se ejecuten de la forma esperada y que se asegure el proceso de calidad del software al momento de la ejecución de pruebas de aceptación. El sistema puede funcionar de manera local, sin embargo, es recomendable desplegar el software en un servidor propio como es Xampp o en la nube para asegurar el correcto funcionamiento de sus módulos y obtener el máximo provecho del sitio web.

IV. REFERENCIAS BIBLIOGRÁFICAS

- AméricaEConomía.com. (2017). La mala gestión de inventarios, una de las principales causas de quiebre de las PyMes. AméricaEConomía.com, 1.
- Cornejo Gómez, D. (marzo de 2017). Universidad politécnica salesiana Ecuador. Obtenido de Análisis de la falta de control en el sistema de inventarios y logística de la empresa Fármaco Veterinaria S.A. Favessa de la ciudad de:
<https://dspace.ups.edu.ec/bitstream/123456789/12631/1/UPS-GT001652.pdf>.
- Sánchez Villalva , D. (2018). ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO. Obtenido de “DISEÑO E IMPLEMENTACIÓN DE UNA APLICACIÓN WEB PARA EL CONTROL DE INVENTARIO DE LA EMPRESA “DGC JEANS”, UTILIZANDO SPRING Y ORACLE.”: <http://dspace.esPOCH.edu.ec/handle/123456789/9102>
- Lema Sagbaycela, F. (Julio de 2018). ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO . Obtenido de “DESARROLLO DEL SISTEMA WEB PARA EL CONTROL DE INVENTARIOS, VENTAS, FACTURACIÓN Y PUBLICIDAD DEL TALLER DE ALUMINIO Y VIDRIO “LÓPEZ” APLICANDO LA METODOLOGÍA LEAN SOFTWARE DEVELOPMENT”:
<http://dspace.esPOCH.edu.ec/handle/123456789/9116>
- Vinueza Aizaga, N. (2018). UNIVERSIDAD TECNOLÓGICA ISRAEL. Obtenido de DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA INFORMÁTICO PARA EL CONTROL DE EXISTENCIAS DE BODEGA CENTRAL DE FARMACIAS.:
<http://repositorio.uisrael.edu.ec/bitstream/47000/1580/1/UISRAEL-EC-SIS-378.242-2018-003.pdf>
- Jaile, M. (2018). FACULTAD DE ESTUDIOS A DISTANCIA 1 UNIDAD 3. GESTIÓN DE INVENTARIOS. Obtenido de http://virtual.umng.edu.co/distancia/ecosistema/ovas/administracion_empresas/logistica/unidad_3/DM.pdf.

- Quizhpi Campoverde, DF (2018). Diseño de un sistema de Control de Inventario y Organización de las Bodegas de producto terminado de la empresa EcuEspumas-Lamitex SA In UNIVERSIDAD POLITÉCNICA SALESIANA SEDE CUENCA CARRERA DE INGENIERÍA INDUSTRIAL. Obtenido de <https://dspace.ups.edu.ec/bitstream/123456789/15717/1/UPS-CT007711.pdf>
- Culqui, J. (2019). Diseño e implementación de un sistema de control, en la ciudad de Chachapoyas, Amazonas. Utp.edu.pe. <https://doi.org/https://hdl.handle.net/20.500.12867/2792>
- Cardona Tunubala, J. L., Orejuela Cabrera, J. P., y Rojas Trejos, C. A. (2018). Gestión de inventario y almacenamiento de materias primas en el sector de alimentos concentrados. Revista EIA, 15(30), 195–208. <https://doi.org/10.24050/reia.v15i30.1066>
- Alava, M (2021). Control interno de los inventarios en la ferreteria ferrisariato hermanos segarra” en la Parroquia Tenguel. Ug.edu.ec. <https://doi.org/http://repositorio.ug.edu.ec/handle/redug/58197>
- Serna, J., Gonzalez, L. J., y Aristizabal, A. F. (2018). SISTEMA DE CONTROL DE INVENTARIO. Tdea.edu.co. <https://doi.org/https://dspace.tdea.edu.co/handle/tda/375>
- Paredes, S (2018). Diseño de manual para controlar inventario de insumos de Agroaereo Fruitexport S.A. Ug.edu.ec. <https://doi.org/http://repositorio.ug.edu.ec/handle/redug/37675>
- Cabello, G. (2019). Modelo de inventarios para ítems con crecimiento, calidad imperfecta y faltantes planeados. Repositorio.tec.mx. <https://doi.org/http://hdl.handle.net/11285/636290>
- López, B., y Isabel, G. (2018). Gestión de inventarios para reducir los costos del almacén de Manpower Perú E.I.R.L. INGnosis, 4(1), 15–28. <https://doi.org/10.18050/ingnosis.v4i1.1572>

- González, A. (2020). Un modelo de gestión de inventarios basado en estrategia competitiva. *Ingeniare. Revista Chilena de Ingeniería*, 28(1), 133–142. <https://doi.org/10.4067/s0718-33052020000100133>
- Batero-Manso, D. F., y Orjuela-Castro, J. A. (2018). El Problema de Ruteo e Inventarios en Cadenas de Suministro de Perecederos: Revisión de Literatura. *Ingeniería*, 23(2). <https://doi.org/10.14483/23448393.12691>
- Hernández, M. (2017). SISTEMA DE CONTROL DE INVENTARIO Obtenido de <https://dspace.tdea.edu.co/bitstream/handle/tda/375/SISTEMA%20DE%20CONTROL%20DE%20INVENTARIO.pdf>
- Méndez, D. (2018). Gestión de Stocks. Universidad Nacional de Mar Del Plata Obtenido de http://nulan.mdp.edu.ar/1830/1/gestion_stock.pdf.
- Elizabeth, M., Marcelo, D., y Ormaza, J. E. (2020). Control y valoración de inventarios con aplicación de NIC 2 en comercializadoras de insumos agrícolas. *Revista Arbitrada Interdisciplinaria Koinonía*, 5(4), 473–503. <https://doi.org/https://dialnet.unirioja.es/descarga/articulo/7651663.pdf>
- Alejos, L. (2019). El uso de software para inventarios en los últimos 10 años. una revisión de la literatura científica. *Upn.edu.pe*. <https://doi.org/https://hdl.handle.net/11537/22559>
- Cerquera, K. (2019). Lectura Método de Inventarios Perpetuos. Obtenido de https://online.ieu.edu.mx/page_Archivos/img_pwtn/2019/07_julio/Inventarios_perpetuos.pdf
- Muñoz, E. (2021). Control y medición de los inventarios en el sector comercial de la provincia de Santa Elena. *Upse.edu.ec*. <https://doi.org/UPSE-TCA-2021-0013>
- Marshall, L. (01 de Noviembre de 2020). Codigos de Barra 2020. *Diccionario electrónico de enseñanza y aprendizaje de lenguas*. Obtenido de Matriz de valoración: <https://www.dicenlen.eu/es/diccionario/entradas/matriz-valoracion>, con fecha de acceso 25/01/2020).

- Amsler, S., y Shea, S. (abril de 2021). RFID o identificación por radiofrecuencia. Obtenido de https://www.madrimasd.org/uploads/informacionidi/biblioteca/publicacion/doc/VT/_VT13_RFID.pdf
- Barlot, M. (2019). El uso de software libre en el control de inventarios: caso de estudio. Obtenido de <https://www.uv.mx/iiesca/files/2019/10/08CA201901.pdf>
- Maida, E., y Pacienza, J. (2017). Biblioteca digital de la Universidad Católica Argentina. Obtenido de Metodologías de desarrollo de software: <https://repositorio.uca.edu.ar/bitstream/123456789/522/1/metodologias-desarrollo-software.pdf>
- Simon, C. (2020). ADAPTACIÓN DE UNA METODOLOGÍA ÁGIL DE DESARROLLO DE SOFTWARE PARA EL BANCO DE OCCIDENTE. Obtenido de <https://red.uao.edu.co/bitstream/handle/10614/11697/T08885.pdf?sequence=4&isAllowed=y>
- Tymkiw, N., Bournissen, J. M., y Tumino, M. C. (2020). SCRUM como Herramienta Metodológica para el Aprendizaje de la Programación. Revista Iberoamericana de Tecnología En Educación Y Educación En Tecnología, (26), 81–89. Obtenido de http://www.scielo.org.ar/scielo.php?pid=S1850-99592020000200010&script=sci_abstract&tlng=en
- González Sajiúm, Derby, y José, R. (2019). Integrando el Scrum a la planificación de proyectos por cadena crítica. Intec.edu.do. <https://doi.org/2636-218X>
- Calvo, D. (2018). UNIVERSIDAD NACIONAL AUTONOMA DE NICARAGUA, (RURD) FACULTAD DE CIENCIAS E INGENIERÍA DEPARTAMENTO DE COMPUTACIÓN TEMA: METODOLOGIA ÁGIL DE DESARROLLO DE SOFTWARE PROGRAMACION. Obtenido de <https://repositorio.unan.edu.ni/1365/1/62161.pdf>.

- Canive, T., y Balet, R. (2020). Reglas y Prácticas en eXtreme Programming. Obtenido de <https://iie.fing.edu.uy/~josej/docs/XP%20-%20Jose%20Joskowicz.pdf>
- Daniela Saavedra Menchaca. (2021). Calidad en la especificación de requerimientos de software aplicado en metodologías ágiles. *INF-FCPN-PGI Revista PGI*, 61–64. Obtenido de https://ojs.umsa.bo/ojs/index.php/inf_fcpn_pgi/article/view/49
- Roux, M. J., Resa, A. L., Higa, R., Macchi, R., y Martín, G. (2020). Eficacia de XP-endo Finisher R y xilol en la remoción de gutapercha del conducto radicular. *Methodo Investigación Aplicada a Las Ciencias Biológicas*, 6(3). [https://doi.org/10.22529/me.2021.6\(3\)06](https://doi.org/10.22529/me.2021.6(3)06)
- Acevedo, J., Lezcano, A., y Irrazábal, E. (2020). Revisión sistemática de la literatura sobre implementación de arquitecturas software para sistemas críticos. *Unlp.edu.ar*. <https://doi.org/http://sedici.unlp.edu.ar/handle/10915/114080>
- Moyano, J. H., Cenci, K. M., y Ardenghi, J. R. (2020). Arquitectura Cliente-Servidor de Alto Rendimiento para servicio RTK. *Unlp.edu.ar*. <https://doi.org/http://sedici.unlp.edu.ar/handle/10915/114094>
- Blanco, J y Rojas, L. (2020). Diseño e implementación de software basado en la arquitectura por capas para la gestión de la cadena de suministro de la empresa de Interventoría Cal y Mayor. *Udistrital.edu.co*. <https://doi.org/http://hdl.handle.net/11349/25712>
- Carlos, R. (2018). Desarrollo de un Sistema para gestionar órdenes de un Restaurante, usando MVC. *Puce.edu.ec*. <https://doi.org/http://repositorio.puce.edu.ec/handle/22000/15842>
- Orozco Aguirre, Héctor Rafael. (2019). UNIDAD DE APRENDIZAJE: LENGUAJE DE PROGRAMACIÓN VISUAL TEMA: MODELO VISTA CONTROLADOR (MVC) Y WEB ARCHIVES (WARS) Ingeniería en Computación. Obtenido de https://ri.uaemex.mx/bitstream/handle/20.500.11799/108712/secme-35486_1.pdf?sequence=1

- Castro-Rivera, V. P., Herrera-Acuña, R. A., y Villalobos-Abarca, M. A. (2020). Desarrollo de un software web para la generación de planes de gestión de riesgos de software. *Información Tecnológica*, 31(3), 135–148. <https://doi.org/10.4067/s0718-07642020000300135>
- Rolando, O. (2017). Diseño e implementación de una arquitectura de software con soporte para dispositivos móviles que apoye la enseñanza musical en niños. *Javeriana.edu.co*. <https://doi.org/http://hdl.handle.net/10554/39888>
- Rivera López, A. (2017). Sistema asistente para la generación de horarios de cursos. Obtenido de *Bibliotecas UDLAP*: http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/rivera_1_a/
- Falcon Mallqui, Samuel. (2018). El lenguaje de programación Java usado en las telecomunicaciones. *Une.edu.pe*. <https://doi.org/M TE-Ti F18 2018>
- Talamé, L., Cardoso, A., y Amor, M. (2019). Comparación de herramientas de procesamiento de textos en español extraídos de una red social para Python. *Unlp.edu.ar*. <https://doi.org/http://sedici.unlp.edu.ar/handle/10915/87854>
- Mina, E., y Cedeño, S. (4 de 4 de 2018). Análisis comparativo entre ASP.NETy PHP. Obtenido de Universidad Ecotec: <http://201.159.222.115/index.php/innova/article/view/474>
- Ovando Ortega, D. J. (2019). UNIVERSIDAD POLITÉCNICA DE SINALOA PROGRAMA ACADÉMICO DE INGENIERÍA EN INFORMÁTICA. In “Bootstrap y Laravel, herramientas para el desarrollo de aplicaciones web.” Obtenido de <http://repositorio.upsin.edu.mx/Fragmentos/tesinas/142016030030OvandoOrtegaDenz elJavier10843.pdf>
- Sunardi, A. (2019). MVC Architecture: A Comparative Study Between Laravel Framework and Slim Framework in Freelancer Project Monitoring System Web Based. *Procedia Computer Science*, 157, 134–141. <https://doi.org/10.1016/j.procs.2019.08.150>

- Muhamad, A. N. (2021). Implementasi Framework Codeigniter V4 Untuk Pengembangan Back-End Website pada Kejaksaan Negeri Kabupaten Pekalongan - Repository IT Telkom Purwokerto. Ittelkom-Pwt.ac.id. <https://doi.org/http://repository.ittelkom-pwt.ac.id/7009/4/cover%20.pdf>
- Méndez, M., Caldevilla, A., Dayanne, B., y Vázquez, C. (2022). Vista de Plataforma para la gestión, administración y monitoreo de servidores PostgreSQL. Obtenido de Revistatonoetecsca.cu [website: http://www.revistatonoetecsca.cu/index.php/tono/article/view/193/168](http://www.revistatonoetecsca.cu/index.php/tono/article/view/193/168)
- Deyimar, A. (2019, December 20). The Comparison and Evaluation of SQLite and MySQL. Obtenido de Academia.edu [website: https://www.academia.edu/35637169/The_Comparison_and_Evaluation_of_SQLite_and_MySQL](https://www.academia.edu/35637169/The_Comparison_and_Evaluation_of_SQLite_and_MySQL)
- Rouse, M. (2017). ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA DE TELECOMUNICACIÓN UNIVERSIDAD POLITÉCNICA DE CARTAGENA Proyecto Fin de Carrera Desarrollo de herramientas web de gestión docente. Obtenida de <https://repositorio.upct.es/bitstream/handle/10317/179/pfc2475.pdf>
- Quinaluiza, A. (2018). Interfaz de programación de aplicaciones para la generación automática de procedimientos almacenados en Mysql. Uta.edu.ec. <https://doi.org/http://repositorio.uta.edu.ec/jspui/handle/123456789/27820>
- Carrizo, D., y Alfaro, A. (2018). Método de aseguramiento de la calidad en una metodología de desarrollo de software: un enfoque práctico. *Ingeniare. Revista Chilena de Ingeniería*, 26(1), 114–129. <https://doi.org/10.4067/s0718-33052018000100114>
- Abrego Almazán, D., Sánchez Tovar, Y., y Medina Quintero, J. M. (2017). Influencia de los sistemas de información en los resultados organizacionales. *Contaduría Y Administración*, 62(2), 303–320. <https://doi.org/10.1016/j.cya.2016.07.005>
- Nicomedes, V. (2017). Tipos de Investigación científica. In *redalyc*. Obtenido de <https://www.redalyc.org/pdf/636/63638739004.pdf>.

Alberto, C. (2020). Los alcances de una investigación. *CienciAmérica: Revista de Divulgación Científica de La Universidad Tecnológica Indoamérica*, 9(3), 1–6. <https://doi.org/https://dialnet.unirioja.es/descarga/articulo/7746475.pdf>

Álvarez Risco, A. (2021). Clasificación de las investigaciones. *Ulima.edu.pe*. <https://doi.org/https://hdl.handle.net/20.500.12724/10818>

Guilln, L. (2017). LA OBSERVACIÒN COMO ESTRATEGIA DE INVESTIGACIÒN PARA CONSTRUIR CONTEXTOS DE APRENDIZAJE Y FOMENTAR PROCESOS PARTICIPATIVOS. 17(1), 201–220. <https://doi.org/10.5944/educxx1.17.1.1074>.

Trindade, V. (14 y 15 de septiembre de 2017). SEDICI. Obtenido de La entrevista no estructurada en investigación cualitativa: una experiencia de campo: <http://sedici.unlp.edu.ar/handle/10915/64407>

Escarcega, D. (2022). La entrevista. Obtenido de <http://diposit.ub.edu/dspace/bitstream/2445/99003/1/entrevista%20pf.pdf>

Palacios Martínez, I. (. (2019). Diccionario electrónico de enseñanza y aprendizaje de lenguas. Obtenido de Matriz de valoración: <https://www.dicenlen.eu/es/diccionario/entradas/matriz-valoracion>, con fecha de acceso 25/01/2019).

Patricia, G., Eduardo, A., y Esther, N. (2020). Metodologías de investigación educativa (descriptivas, experimentales, participativas, y de investigación-acción). *RECIMUNDO*, 4(3), 163–173. [https://doi.org/10.26820/recimundo/4.\(3\).julio.2020.163-173](https://doi.org/10.26820/recimundo/4.(3).julio.2020.163-173)

V. ANEXOS

Anexo 1. Certificado o acta del perfil de investigación



UNIVERSIDAD POLITÉCNICA ESTATAL DEL CARCHI
FACULTAD DE INDUSTRIAS AGROPECUARIAS Y CIENCIAS AMBIENTALES
CARRERA DE INGENIERÍA EN INFORMÁTICA

ACTA

DE LA SUSTENTACIÓN DE PREDEFENSA DEL INFORME DE INVESTIGACIÓN DE:

NOMBRE: Sr. Darwin Fernando Tipan Viveros **CÉDULA DE IDENTIDAD:** 1725098899
NIVEL/PARALELO: 0 **PERIODO ACADÉMICO:** PAO 2021B

TEMA DE INVESTIGACIÓN: Propuesta de un sistema de gestión de inventarios para la farmacia del Hospital del Día Nova Salud

Tribunal designado por la dirección de esta Carrera, conformado por:

PRESIDENTE: Msc. Marco Antonio Yandún Velasteguí
LECTOR: MSc. Jeffery Alex Naranjo Cedeño
ASESOR: Msc. Samuel Benjamin Lascano Rivera

De acuerdo al artículo 21: Una vez entregados los requisitos para la realización de la pre-defensa el Director de Carrera integrará el Tribunal de Pre-defensa del informe de investigación, fijando lugar, fecha y hora para la realización de este acto:

EDIFICIO DE AULAS: 4 **AULA:** 209
FECHA: 2 de marzo de 2022
HORA: 0,375

Obteniendo las siguientes notas:

1) Sustentación de la predefensa: 5,85
2) Trabajo escrito 2,80
Nota final de PRE DEFENSA 8,65

Por lo tanto: **APRUEBA CON OBSERVACIONES** ; debiendo acatar el siguiente artículo:

Art. 24.- De los estudiantes que aprueban el Plan de Investigación con observaciones. - El estudiante tendrá el plazo de 10 días laborables para proceder a corregir su Informe de Investigación de conformidad a las observaciones y recomendaciones realizadas por los miembros Tribunal de sustentación de la pre-defensa.

Para constancia del presente, firman en la ciudad de Tulcán el 2 de marzo de 2022



Marco Antonio
YANDÚN
VELASTEGUÍ

Msc. Marco Antonio Yandún Velasteguí

PRESIDENTE



Samuel Benjamin
LASCANO RIVERA

Msc. Samuel Benjamin Lascano Rivera

TUTOR



Jeffery Alex
NARANJO CEDEÑO

MSc. Jeffery Alex Naranjo Cedeño

LECTOR

Adj.: Observaciones y recomendaciones

Anexo 2. Certificado del abstract por parte del centro de idiomas



**UNIVERSIDAD POLITÉCNICA ESTATAL DEL CARCHI
FOREIGN AND NATIVE LANGUAGE CENTER**

ABSTRACT- EVALUATION SHEET				
NAME: Tipán Viveros Darwin Fernando				
DATE: 10 de marzo de 2022				
TOPIC: "PROPUESTA DE UN SISTEMA DE GESTIÓN DE INVENTARIOS PARA LA FARMACIA DEL HOSPITAL DEL DIA NOVA SALUD"				
MARKS AWARDED QUANTITATIVE AND QUALITATIVE				
VOCABULARY AND WORD USE	Use new learnt vocabulary and precise words related to the topic	Use a little new vocabulary and some appropriate words related to the topic	Use basic vocabulary and simplistic words related to the topic	Limited vocabulary and inadequate words related to the topic
	EXCELLENT: 2 <input checked="" type="checkbox"/>	GOOD: 1,5 <input type="checkbox"/>	AVERAGE: 1 <input type="checkbox"/>	LIMITED: 0,5 <input type="checkbox"/>
WRITING COHESION	Clear and logical progression of ideas and supporting paragraphs. <input checked="" type="checkbox"/>	Adequate progression of ideas and supporting paragraphs. <input type="checkbox"/>	Some progression of ideas and supporting paragraphs. <input type="checkbox"/>	Inadequate ideas and supporting paragraphs. <input type="checkbox"/>
	EXCELLENT: 2 <input checked="" type="checkbox"/>	GOOD: 1,5 <input type="checkbox"/>	AVERAGE: 1 <input type="checkbox"/>	LIMITED: 0,5 <input type="checkbox"/>
ARGUMENT	The message has been communicated very well and identify the type of text <input checked="" type="checkbox"/>	The message has been communicated appropriately and identify the type of text <input type="checkbox"/>	Some of the message has been communicated and the type of text is little confusing <input type="checkbox"/>	The message hasn't been communicated and the type of text is inadequate <input type="checkbox"/>
	EXCELLENT: 2 <input checked="" type="checkbox"/>	GOOD: 1,5 <input type="checkbox"/>	AVERAGE: 1 <input type="checkbox"/>	LIMITED: 0,5 <input type="checkbox"/>
CREATIVITY	Outstanding flow of ideas and events <input type="checkbox"/>	Good flow of ideas and events <input checked="" type="checkbox"/>	Average flow of ideas and events <input type="checkbox"/>	Poor flow of ideas and events <input type="checkbox"/>
	EXCELLENT: 2 <input type="checkbox"/>	GOOD: 1,5 <input checked="" type="checkbox"/>	AVERAGE: 1 <input type="checkbox"/>	LIMITED: 0,5 <input type="checkbox"/>
SCIENTIFIC SUSTAINABILITY	Reasonable, specific and supportable opinion or thesis statement <input checked="" type="checkbox"/>	Minor errors when supporting the thesis statement <input type="checkbox"/>	Some errors when supporting the thesis statement <input type="checkbox"/>	Lots of errors when supporting the thesis statement <input type="checkbox"/>
	EXCELLENT: 2 <input checked="" type="checkbox"/>	GOOD: 1,5 <input type="checkbox"/>	AVERAGE: 1 <input type="checkbox"/>	LIMITED: 0,5 <input type="checkbox"/>
TOTAL/AVERAGE	9 - 10: EXCELLENT 7 - 8,9: GOOD 5 - 6,9: AVERAGE 0 - 4,9: LIMITED		TOTAL 9	



**UNIVERSIDAD POLITÉCNICA ESTATAL DEL
CARCHI FOREIGN AND NATIVE LANGUAGE
CENTER**

Informe sobre el Abstract de Artículo Científico o Investigación.

Autor: Tipán Viveros Darwin Fernando

Fecha de recepción del abstract: 10 de marzo de 2022

Fecha de entrega del informe: 10 de marzo de 2022

El presente informe validará la traducción del idioma español al inglés si alcanza un porcentaje de: 9 – 10 Excelente.

Si la traducción no está dentro de los parámetros de 9 – 10, el autor deberá realizar las observaciones presentadas en el ABSTRACT, para su posterior presentación y aprobación.

Observaciones:

Después de realizar la revisión del presente abstract, éste presenta una apropiada traducción sobre el tema planteado en el idioma Inglés. Según los rubrics de evaluación de la traducción en Inglés, ésta alcanza un valor de 9 por lo cual se valida dicho trabajo.

Atentamente



EDISON SOAMES
PENAFIEL ARCOS

Ing. Edison Peñafiel Arcos MSc
Coordinador del CIDEN

Anexo 3. Certificado de plagio turnitin

TESIS DE GRADO

INFORME DE ORIGINALIDAD

3%	%	3%	%
INDICE DE SIMILITUD	FUENTES DE INTERNET	PUBLICACIONES	TRABAJOS DEL ESTUDIANTE

FUENTES PRIMARIAS

1	UNIVERSIDAD ESTATAL DEL SUR DE MANABÍ. "VI CONGRESO INTERNACIONAL DE INGENIERÍAS: "INGENIERÍA PARA FORMAR UNA SOCIEDAD SOSTENIBLE"", Editorial Internacional Runaiki, 2019 Publicación	1%
2	Griselda Cortes, Alicia Valdez, Laura Vazquez, Alma Dominguez, Cesar Gonzalez, Ernestina Leija, Jose Cendejas. "Citizen Attention Web Application for the Municipality of Sabinas, Coahila, Mexico", International Journal of Advanced Computer Science and Applications, 2019 Publicación	<1%
3	Enrique Zambrano-Pilay, Sergio Carreño-Lucas, Edison Almeida-Zambrano. "Desarrollo e implementación del sistema de gestión editorial de la ULEAM", Revista Científica INGENIAR: Ingeniería, Tecnología e Investigación, 2020 Publicación	<1%

Anexo 4. Autorización de recolección de información para la realización del proyecto

UPEC-CC-2019-247-OF
Tulcán, 02 de septiembre de 2019

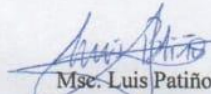
DOCTOR
Carlos Pozo
GERENTE DEL HOSPITAL DEL DÍA NOVA SALUD
Presente.-

De mi consideración:

En mi calidad de Director de la Carrera de Computación e Ingeniería en Informática, de la Universidad Politécnica Estatal del Carchi, me dirijo a usted para solicitarle de la manera más comedida nos facilite información referente al inventario de farmacia, misma que será utilizada para desarrollar el tema del perfil de proyecto de titulación por parte del estudiante Tipan Viveros Darwin Fernando con número de cédula 1725098899.

Por la atención que se digne dar al presente, reciba mis agradecimientos.

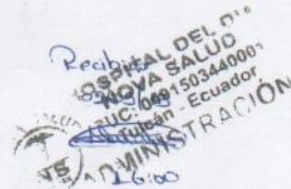
Atentamente,


Msc. Luis Patiño



DIRECTOR CARRERA DE COMPUTACIÓN
INGENIERÍA EN INFORMÁTICA
“EDUCACIÓN PARA EL DESARROLLO Y LA INTEGRACIÓN”
C.I. 0401063870 Telf.: 2224079 Ext. 2030

LP/vm



Anexo 5. Certificado de culminación del proyecto de investigación



“CENTRO MEDICO QUIRURGICO AMBULATORIO
HOSPITAL DEL DIA NOVA SALUD”
UN COMPROMISO PERMANENTE CON SU SALUD
Dirección: Av. El Centenario y Av. Veintimilla
Telfs: 2 984 467
TULCAN - ECUADOR

Tulcán, 06 de septiembre del 2021

A petición del interesado

CERTIFICADO DE CONFORMIDAD

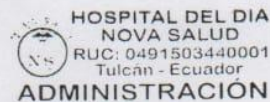
Certifico que el Sr. **Darwin Fernando Tipan Viveros** portador de la cédula de identidad No. 172509889-9 elaboro el software con las especificaciones que se le dio al principio, de tal manera que el Software diseñado nos permite el control de inventario que es el modulo principal con módulos adicionales que en su desarrollo fueron adaptados a los procesos que se le indico.

Por lo que dicho software cumple con las expectativas de la Institución y no necesita ser modificado, por lo que la Institución agradece al Sr. Por la elaboración de dicho software que será de gran ayuda para el manejo del Inventario en nuestra Institución.

Es todo cuanto puedo certificar, faculto al interesado hacer uso del presente en los casos que estime conveniente.

Atentamente

Lic. Carmen Zúñiga Rosero
ADMINISTRADORA



Anexo 6. Entrevista a la señorita administradora y al señor bodeguero



Universidad politécnica estatal del Carchi
Facultad de Industrias Agropecuarias y Ciencias Ambientales
Carrera de Ingeniería en Informática

Formación de entrevistas no estructurados para la parte Administrativa del Hospital del Día Nova Salud

La presente entrevista recolectó información relacionada con el procesó de inventario la cual tiene el objetivo en implementar los procesos funcionales en el sistema.

Preguntas	Entrevistado	Respuesta
1. ¿A nivel general qué procesos se realizan en la clínica?	Administradora	
2. ¿De qué forma se realizan los procesos administrativos sobre el control de inventario en el área de farmacia?	Administradora	
3. ¿Al no tener un control exacto de sus materiales y accesorios en bodega, como se lo lleva actualmente?	Administradora	
4 ¿Cuál es el problema de no contar con un registro exacto de los materiales y accesorios en el área de farmacia?	Administradora	
5. ¿De qué manera se realiza el detalle de los precios de venta al público?	Administradora	
6. ¿Cómo se realiza la compra de recetas de pacientes internos?	Administradora	
7. ¿Cómo se determina la fecha de caducidad de los materiales y accesorios?	Administradora	
8. ¿Qué tipo de inventario se utiliza en el área de farmacia? ¿Para su mayor comprensión véase la siguiente información?	Administradora	



**Formación de entrevistas no estructurados para el encargado de bodega del
 Hospital del Día Nova Salud**

La presente entrevista recolectó información relacionada con el procesó del sistema del inventario.

Preguntas	Entrevistado	Respuesta
1. ¿Tiene acceso a internet la bodega y las áreas respectivas como despacho y recibimiento de productos?	Bodeguero	
2. ¿Usted tiene conocimiento y dominio de los tipos de inventario que existen como son los tres tipos ABC? Véase la siguiente información	Bodeguero	

Tipos de Inventario		
A	B	C
Productos de alto valor, de gran venta, que requieren de mayor atención y cuidado a través de:	Productos de alto valor con ventas moderadas, requiere un tratamiento normal; es decir una atención ajustada a los requerimientos del negocio	Productos de bajo valor y poca venta, que deben tratarse según el principio de la simplificación productiva y administrativa y de reducción de costos.
Análisis de mercado, de precios y de costos		Requerimientos simplificados de inventarios
Registro y control de inventarios		Trámites simplificados en el manejo de pedidos y pedidos de grandes cantidades
Determinación precisa de seguridad		Supervisión de las existencias
Aplicación preferencial del análisis de valores		

Preguntas	Entrevistado	Respuesta
3. ¿En la escala del 1 al 10% qué conocimiento tiene usted acerca de computación?	Bodeguero	
4. ¿Cuáles son los pasos que usted realiza al momento de recibir un pedido?	Bodeguero	
5. ¿Cómo es el manejo del control de existencia en los materiales y accesorios que se encuentran en farmacia?	Bodeguero	
6. ¿Cómo se controla la entrada o salida de los materiales y accesorios adquiridos?	Bodeguero	
7. ¿Qué método se utiliza Fifo/Lifo para los materiales y accesorios?	Bodeguero	

Anexo 7. Tabla comparativa de metodologías ágiles por, MÉNDEZ, E. R. (2018, July). ESTIMACIÓN DE ESFUERZO EN PROYECTOS DE DESARROLLO DE SOFTWARE CON METODOLOGÍAS ÁGILES. Obtenido de Universidad Politécnica de Valencia website: https://riunet.upv.es/bitstream/handle/10251/109927/PSC9691917_TFM_1532552620188720975771735607734.pdf?sequence=1&isAllowed=y.

Metodologías	Referencias principales	Tamaño del proyecto	Tamaño del equipo	Modelo de desarrollo	Distribución física	Cultura de negocio
Adaptive Software Development	(Highsmith, 2000)	Grandes y complejos	No especificado	Desarrollo iterativo, rápido y distribuido	Equipos coubicados y distribuidos	No especificada
Crystal	(Cockburn, 2002)	Pequeños y medianos	Un equipo, máximo de 6 personas (Crystal Clear); múltiples equipos: máximo de 40 personas (Crystal Orange), entre 40 y 80 personas (Crystal Red)	Desarrollo iterativo y rápido	Equipos coubicados (sin soporte para desarrollo distribuido)	No especificada
Dynamic Systems Development Method (DSDM)	(Stapleton, 1997)	Pequeños y grandes (aplicaciones de negocio)	Mínimo 2 y máximo 6 personas (múltiples equipos)	Desarrollo iterativo, rápido y cooperativo	No especificada	Colaborativa y cooperativa

Tabla 2.1: Metodologías de desarrollo ágil. Adaptado de (Qumer y Henderson-Sellers, 2008). [cont.]

Metodologías	Referencias principales	Tamaño del proyecto	Tamaño del equipo	Modelo de desarrollo	Distribución física	Cultura de negocio
Extreme Programming (XP)	(Beck, 2000)	Pequeños y medianos	Máximo 10 personas	Desarrollo iterativo y rápido	Equipos coubicados y distribuidos (iteraciones limitadas)	Colaborativa y cooperativa
Feature Driven Development (FDD)	(Palmer & Felsing, 2002)	Pequeños, medianos y grandes (proyectos y aplicaciones de negocio)	Sin límites (escalable desde pequeños hasta grandes equipos)	Diseño y construcción iterativos	No especificada	No especificada
Lean Software Development	(M. B. Poppendieck & Poppendieck, 2003)	Grandes y complejos	Hasta 50 personas	Desarrollo iterativo y rápido	Equipos coubicados y distribuidos	Basada en los principios de "Lean Thinking"
Scrum	(Schwaber, 1995; Schwaber & Beedle, 2002)	Pequeños y medianos (escalable para grandes proyectos)	Máximo 10 personas (múltiples equipos)	Desarrollo iterativo y rápido	No especificada	No especificada

Imagen de tabla de los aspectos claves y deficiencias de Adaptativa Software Development.

Aspectos clave (+)	Principales deficiencias (-)
Colaboración; Desarrollo iterativo; Organizaciones vistas como sistemas adaptativos.	Comúnmente considerado como modelo conceptual de cultura organizacional más que como una práctica de desarrollo de software.

Imagen de tabla de los aspectos clave y deficiencias de Xp.

Aspectos clave (+)	Principales deficiencias (-)
Desarrollo orientado al cliente; Implementación de equipos pequeños; Búsqueda de excelencia técnica; Promoción del uso de estándares de codificación.	Se ofrece menos atención a la supervisión y gestión general, y más al desempeño individual. Cualquier resistencia frente a los principios y prácticas generales de XP, tanto por parte de los miembros del equipo como de los clientes, puede provocar que falle todo el proceso.

Imagen de tabla de los aspectos clave y deficiencias de Scrum.

Aspectos clave (+)	Principales deficiencias (-)
Equipos independientes, pequeños, multifuncionales y auto-organizables; Seguimiento diario (<i>stand-up meetings</i>); Enfoque a gestión del proyecto.	Las fases de integración y prueba no aparecen detalladas; Se recomienda su implementación con equipos de más de 10 personas.

Imagen de tabla de mapeo entre ASD y las áreas de conocimiento del PMBoK. Aquí se puede evidenciar algunas de sus características para llevarlas a la tabla comparativa.

Áreas de conocimiento del PMBoK	Principios de desarrollo de software ágil	Mejores prácticas en ASD
Gestión de la integración	Integración continua	<ul style="list-style-type: none"> • Tratar una interfaz de usuario como un contrato entre el equipo de desarrollo y el sistema. • Construir y probar las interfaces de usuario en las primeras etapas del proyecto, gestionando a tiempo los puntos de riesgo del proyecto. • Construir emuladores para probar el funcionamiento del sistema como está destinado a ser utilizado.
Gestión del alcance	Aceptación del cambio	<ul style="list-style-type: none"> • Utilizar iteraciones rápidas para desarrollar software que funcione y del que se pueda recibir retroalimentación. • Planificar iteraciones cortas, reduciendo la incertidumbre y descomponiendo el trabajo en pequeños entregables. • Programar demostraciones del funcionamiento del software a los clientes al final de cada iteración.
Gestión del tiempo	Estimación y gestión del tiempo	<ul style="list-style-type: none"> • Utilizar el <i>timeboxing</i> en la programación de la duración de las actividades del proyecto. • Estimar el trabajo a realizar utilizando puntos de historia (<i>story points</i>), así como también evaluar y mejorar los resultados de las estimaciones realizadas en cada iteración. • Considerar el trabajo completado en iteraciones pasadas como medida máxima de trabajo estimado para nuevas iteraciones.
Gestión de los costos	Entrega continua	<ul style="list-style-type: none"> • Definir una lista de criterios y condiciones que el software ha de cumplir para ser considerado como <i>Done</i> (o Listo). • Elaborar escenarios de prueba <i>end-to-end</i>, en los que las partes interesadas puedan determinar en cualquier momento cuando una funcionalidad ha sido implementada correctamente o dentro de los criterios de aceptación.

Tabla 2.9: Mapeo entre ASD y las áreas de conocimiento del PMBoK.
[cont.]

Áreas de conocimiento del PMBoK	Principios de desarrollo de software ágil	Mejores prácticas en ASD
Gestión de la calidad	TDD y automatización de las pruebas	<ul style="list-style-type: none"> • Utilizar TDD como enfoque de pruebas de desarrollo. • Automatizar los casos de prueba, en especial los de más alta prioridad. • Implementar la práctica de integración continua (<i>continuous integration</i>), garantizando que los cambios más recientes no hayan desestabilizado el funcionamiento del proyecto desarrollado y que los mismos cumplen con los criterios de calidad requeridos.
Gestión de los recursos humanos	Liderazgo y equipos auto-gestionados	<ul style="list-style-type: none"> • Contar con equipos que sean responsable de sí mismos. • Asegurar que cada iteración cuenta con reuniones retrospectivas de las que se pueda obtener resultados significativos y accionables. • Reconocer las fortalezas del equipo y desarrollar un ambiente donde la retroalimentación crítica y honesta es valorada.
Gestión de las comunicaciones	Comunicación efectiva y equipos motivados	<ul style="list-style-type: none"> • Hacer uso de reuniones cortas para identificar el progreso del trabajo y los puntos de bloqueo que obstaculicen el avance del proyecto. • Demostrar el progreso del proyecto constantemente a las partes interesadas. • Contar con sesiones de planificación que permitan realizar la priorización de las funcionalidades próximas a ser abordadas, en las que participen cada una de los roles involucrados en el proyecto.
Gestión de los riesgos	Mitigación de los riesgos	<ul style="list-style-type: none"> • Trabajar sobre los elementos de mayor riesgo en las etapas tempranas del proyecto. • Identificar y mitigar los riesgos antes de que se conviertan en problemas reales para el proyecto, mediante prácticas como <i>continuous integration</i> y las reuniones retrospectivas.

Tabla 2.9: Mapeo entre ASD y las áreas de conocimiento del PMBoK.
[cont.]

Áreas de conocimiento del PMBoK	Principios de desarrollo de software ágil	Mejores prácticas en ASD
Gestión de las adquisiciones	Gestión ágil de los contratos	<ul style="list-style-type: none"> • Garantizar que el uso de las buenas prácticas de desarrollo ágil se encuentre estipulado en el contrato y en la definición de los entregables. • Enfocarse en la buena comunicación, la integración y los modelos de prueba para mantener al equipo alineado con los entregables establecidos.
Gestión de las partes interesadas	Trabajo colaborativo	<ul style="list-style-type: none"> • Considerar al usuario final como el principal <i>stakeholder</i> del proyecto. • Garantizar la participación activa de cada uno de los interesados del proyecto. • Promover una cultura de honestidad y confianza.

Imagen de tabla comparativa de las metodologías ágiles por, Porras Vásconez, E. J. (2021). METODOLOGÍA ASD APLICADO AL DESARROLLO DEL SISTEMA DE DIAGNÓSTICO DE SERVICIOS TURÍSTICOS COMUNITARIOS DE LAS PARROQUIAS RURALES DEL CANTÓN RIOBAMBA. Obtenido de Universidad Nacional De Chimborazo. website: <http://dspace.unach.edu.ec/bitstream/51000/8489/3/8%20-%20Informe%20de%20Tesis%20Joaquin%20Porras.pdf>.

Metodologías	Tamaño del proyecto	Tamaño del equipo	Modelo de desarrollo	Distribución física
Adaptative Software Development (ASD)	Grandes y complejos	No especificado	Desarrollo iterativo, rápido y distribuido	Equipos coubicados* y distribuidos
Crystal	Pequeños y medianos	Crystal Clear: máximo de 6; Crystal Orange: múltiples equipos máximo de 40; Crystal Red: entre 40 y 80	Desarrollo iterativo y rápido	Equipos coubicados sin soporte para desarrollo distribuido.
Extreme Programming (XP)	Pequeños y medianos	Máximo 10 personas	Desarrollo iterativo y rápido	Equipos coubicados y distribuidos con interacciones limitadas
Scrum	Pequeños y medianos	De 6 a 9 personas (multifuncionales) (V. E. Guamán Palate, 2020)	Desarrollo iterativo y rápido	No especificado

Imagen de las ventajas y desventajas de las dos metodologías ágiles Xp y Scrum por. Sangama Oñate, AF (2020). Metodologías ágiles Scrum, XP, SLeSS, Scrumban, HME, Mobile-D y MASAN empleadas en la industria de dispositivos móviles: Un contraste en favor de la industria del desarrollo móvil. Obtenido del sitio web de la Universidad Peruana Unión: https://repositorio.upeu.edu.pe/bitstream/handle/20.500.12840/3906/Abel_Trabajo_Bachillerato_2020.pdf?sequence=1&isAllowed=y.

Cuadro comparativo y de contraste de las metodologías más conocidas en el desarrollo de aplicativos móviles	
Practica ágil	Scrum
Autor(es)	En 1993: Jeff Sutherland, John Scumniotales y Jeff McKenna (<i>ejecutaron y documentaron el primer Scrum</i>). En 1995: Ken Schwaber (formalizó el proceso para la industria de desarrollo de software).
Ventajas	Desventajas
<ul style="list-style-type: none"> • Scrum proporciona un enfoque ágil para la gestión de proyectos aumentando la probabilidad de su éxito. • Resultados significativos en desenvolvimiento. • Mayor intercambio de información • Centrarse en cumplir los objetivos y metas del proyecto • Identificación clara de las debilidades del equipo. • Identificación diaria de impedimentos y riesgos. • Claridad de los objetivos del proyecto. • Mejor comunicación y reuniones más rápidas y objetivas. • Mejor distribución del trabajo • Mejor proactividad, motivación, integración y compromiso del equipo. • Monitoreo y control descentralizado por parte del equipo. • Planificación del equipo. 	<ul style="list-style-type: none"> • Scrum puede decirle al equipo que el código debe probarse bien, pero no especifica cómo se realiza esa prueba. • Requiere una exhaustiva definición de las tareas y sus plazos. Cuando estos dos aspectos no se definen adecuadamente, Scrum se desvanece. Recuerda que la división del trabajo en cada etapa (y de éstas en tareas específicas) son la esencia de esta metodología. • Exige que quienes la utilicen cuenten con una alta cualificación o formación. No es una modalidad de gestión propia de grupos junior o que apenas estén en proceso de formación. Gran parte del éxito de Scrum radica en la experiencia que aportan los profesionales de los equipos, quienes por lo general acumulan años de experiencia
Practica ágil	eXtreme Pro- gramming (XP)
Autor(es)	En 1999: Kent Beck
Ventajas	Desventajas
<ul style="list-style-type: none"> • Resultados significativos en desenvolvimiento en algunos proyectos. • XP especificaría el desarrollo impulsado por pruebas (TDD), una receta de cómo el equipo desarrolla el código mediante pruebas; TDD 	<ul style="list-style-type: none"> • XP como una serie de piezas de rompecabezas que encajan como un reloj suizo, a pesar del hecho de que XP es supuestamente adaptable a una amplia variedad de proyectos. Algunas prácticas, aunque no agregan valor, son imposibles de eliminar, ya que son necesarias para mantener las otras en su lugar.

Anexo 8. Imagen de tabla comparativa de las arquitecturas de software por, Alfonso, E. (2020). Desarrollo de una aplicación web con el Framework Bootstrap y el precompilador Sass para la gestión de pedidos de productos agrícolas de la Empresa El Chagra”. Epoch.edu.ec. Obtenido de <http://dspace.epoch.edu.ec/bitstream/123456789/13654/1/18T00807.pdf>.

PARÁMETROS	MONOLÍTICA	CLIENTE-SERVIDOR	N CAPAS
Escalabilidad	NO	SI	SI
Facilidad de mantenimiento	NO	SI	SI
Flexibilidad	NO	SI	SI
Versatilidad	NO	SI	SI
Costos	Altos	Medio	Alto

Imagen de tabla de ventajas y desventajas de la arquitectura Cliente-Servidor.

Ventajas	Desventajas
<ul style="list-style-type: none"> • Facilidad en la integración de los diferentes sistemas, permitiendo compartir información. • Permite al usuario tener mayor interacción con la aplicación. • Genera un orden de trabajo, donde cada cliente puede acceder al mismo servidor sin generar conflictos. • Facilita la integración de nuevas funcionalidades, favoreciendo la estabilidad de las aplicaciones web. 	<ul style="list-style-type: none"> • Necesaria habilidad para reparar un servidor, sin perder la información y el correcto funcionamiento. • En la seguridad se requiere pasar por procesos de validación, mediante protocolos.

Imagen de tabla comparativa entre patrones de diseño.

PARÁMETROS	MVC	PIZARRA	FILTRO DE TUBERÍA
Escalabilidad	SI	NO	SI
Control	Los recursos del servidor	Depende del estado del pizarrón	VARIABLES de transformación
Solución	A través del análisis de la situación real	No existe garantía	Mediante un flujo a través de conectores
Mantenibilidad	Fácil	Difícil	Difícil
Reusabilidad	SI	NO	SI
Seguridad	SI	NO	SI
Explicaciones	SI	NO	NO

Imagen de tabla de las ventajas y desventajas del patrón MVC.

VENTAJAS	DESVENTAJAS
<ul style="list-style-type: none">• Desarrollo en menos tiempo y reutilización de las partes de forma independiente.• En cuanto a la migración cubriendo las altas demandas y permitir agregar nuevos componentes o reemplazar otros.• Facilita el manejo de los errores.	<ul style="list-style-type: none">• En cuanto a la cantidad de archivos en mantener y desarrollar se incrementa.• La separación entre capas resulta en dificultad en el desarrollo del sistema.

Anexo 9. Tabla comparativa de los lenguajes de programación por, Valarezo Pardo, M., Honores Tapia, J., Gómez Moreno, A., y Vines Sánchez, L. (07 de Marzo de 2018). COMPARACIÓN DE TENDENCIAS TECNOLÓGICAS EN APLICACIONES WEB. Obtenido de Universidad Técnica de Machala: https://www.3ciencias.com/wp-content/uploads/2018/09/Art_2.pdf.

Concepto	PHP	Java	Python	Ruby
Costo de servidor	Gratuito	Gratuito	Gratuito	Gratuito
Sintaxis de lenguaje base	C / C++	C/ C++	C/ C++	Perl, Smalltalk, Eiffel, Ada, y Lisp
Orientado a objetos	No completamente	Si	Si	Si
Sistemas operativos	Linux o Windows	Linux o Windows	Linux o Windows	Linux o Windows
Servidor	Apache, compilador propio	Apache, Tomcat y Glassfish	Apache, compilador propio	Apache, compilador propio
Empresa	The PHP Group (Open source)	Oracle (Open source)	Python software foundation(Open source)	Grupo Ruby (Open source)
Base de datos (principalmente)	MySQL	Oracle, MySQL	MySQL y PostgreSQL	MySQL y PostgreSQL
Propósito	Generar dinámicamente páginas Web	Generar dinámicamente páginas Web	Enfatiza la productividad y la lectura fácil del código	Código "divertido" y fácil de modificar por parte del desarrollador.
Apoyo de aprendizaje	Mucha, pero descentralizada. No hay una entidad que de forma oficial centralice la ayuda	Mucha, pero descentralizada. No hay una entidad que de forma oficial centralice la ayuda	Mucha, pero descentralizada. No hay una entidad que de forma oficial centralice la ayuda	Menos, pero descentralizada. No hay una entidad que de forma oficial centralice la ayuda
Soporte a móviles (todos por medio de un browser)		Native: android		
Ambiente de desarrollo	Eclipse y otras herramientas Open Source	Eclipse, Netbeans y otras herramientas Open Source	Eclipse, Netbeans y otras herramientas open Source	Eclipse, Netbeans y otras herramientas Open Source
Soporte a móviles (todos por medio de un browser)		Native: android		
Ambiente de desarrollo	Eclipse y otras herramientas Open Source	Eclipse, Netbeans y otras herramientas Open Source	Eclipse, Netbeans y otras herramientas open Source	Eclipse, Netbeans y otras herramientas Open Source

Imagen de tabla comparativa sobre las características de desarrollo por, Bolívar, S., y Alexandra. (2019). Plataforma tecnológica para contribuir a la planeación urbana de la ciudad de Guayaquil dirigido a la transportación, enfocado a la administración de base de datos Postgresql garantizando la disponibilidad 24/7 de la información, desarrollando esquemas de respaldos automáticos, monitoreo, depuración, seguridad y sincronización. Ug.edu.ec. <https://doi.org/http://repositorio.ug.edu.ec/handle/redug/39980>.

PHP	JAVA	PYTHON	RUBY
Servidor gratuito	Servidor gratuito	Servidor gratuito	Servidor gratuito
Orientado a objetos	Orientado a objetos	Orientado a objetos	Orientado a objetos
Linux/Windows	Linux/Windows	Linux/windows	Linux/windows
Compilador propio Apache	Apache, Tomcat, Glassfish	Compilador propio Apache	Compilador propio Apache
Páginas web dinámicas	Páginas web dinámicas	Lectura fácil de código	Fácil de modificar
Ambiente de desarrollo Eclipse y herramientas open source	Eclipse, Netbeans y otras herramientas open source	Eclipse, Netbeans y otras herramientas open source	Eclipse, Netbeans y otras herramientas open source

Anexo 10. Imágenes de tablas comparativas de los frameworks de desarrollo web por, Sierra, F., Acosta, J., Ariza, J., y Salas, M. (2017). Estudio y análisis de los framework en php basados en el modelo vista controlador para el desarrollo de software orientado a la web. Investigación Y Desarrollo En TIC, 4(2), 14–26. Obtenido de <http://revistas.unisimon.edu.co/index.php/identific/article/view/2480>.

Nombre	Ventajas
Kumbia PHP	<ul style="list-style-type: none"> • Implementa patrones de programación orientados a la web. • Utilización de la web en el software. • Hace la mayor parte del trabajo y se ocupa de los detalles. • Mantenibilidad de la aplicación. • Curva de aprendizaje mínima. • Su estructura de directorio favorece la productividad
Wasp	<ul style="list-style-type: none"> • WASP fue escrito desde cero en objeto puro Oriented PHP5. WASP utiliza al máximo todas las mejoras OO hecho de PHP en la versión 5 incluye encapsulación pública / protected / privada, clases abstractas e interfaces, carga automática de clases, y el manejo de excepciones. □
CodeIgniter	<ul style="list-style-type: none"> • Ofrece un marco con una pequeña huella. • Tiene un excelente rendimiento. • Documentación completa. • Ofrece una amplia contabilidad con alojamiento estándar. • Casi que utiliza cero configuración. • Pasar más tiempo lejos de la computadora. • Soluciones simples

<p>Seagull</p>	<ul style="list-style-type: none"> • Extremadamente fácil de usar. • Ideal para principiantes y expertos. • Para los principiantes Seagull posee una librería con aplicaciones de muestra que pueden ser personalizadas para adaptarse a sus necesidades. • Para los más expertos ofrece muchas opciones para construir aplicaciones web rápida y fácilmente. • Seagull posee una comunidad de desarrolladores activa y mucha documentación de soporte. • Permite realizar una programación modular. • Posee un sistema de Gestión de Contenidos (CMS).
<p>BlueShoes</p>	<ul style="list-style-type: none"> • BlueShoes ofrece un excelente soporte base de datos MySQL. • Soporte para Oracle y MSSQL. • BlueShoes proporciona a los desarrolladores de soluciones del mundo real para los componentes del sistemas comunes. • Gestión de usuarios. <input type="checkbox"/> Gestión de la sesión. <input type="checkbox"/> Cerrado de usuarios. <input type="checkbox"/> Control de excepciones. • Registro de objetos persistente

Qcodo	<ul style="list-style-type: none"> • Qcodo permite que algunos de los sitios web con más tráfico de datos y aplicaciones intensivas para empresas sean de mejor manejo para el usuario. • Marco orientado a objetos. • Toma lo mejor de PHP. • Proporciona una plataforma de
Diy	<ul style="list-style-type: none"> • orientado a objetos y diseñado siguiendo la arquitectura MVC. • Integrar de una forma sencilla y eficaz las tecnologías que ya han sido probadas. • Explota el uso de PHP5 tomándolo por ser el modelo de objetos más avanzado.
Wact	<ul style="list-style-type: none"> • Se desarrolla con la filosofía de la refactorización continua. • Se desarrolla con pruebas unitarias
Zend Framework	No es fácil de aprender, necesita de mucha configuración antes de usarse, estructura de archivos no definida
Yii	Se genera código basura y en ocasiones no lo filtra de la mejor forma y complica el uso de este
CakePHP	Las versiones del framework influyen en las versiones de uso de PHP
Laravel	Es relativamente nuevo y muchos dejan de utilizarlo porque creen que no es confiable
Symfony	No es muy robusto

Nombre Del Framework	Características
Kumbia PHP	<ul style="list-style-type: none"> • Define una filosofía de Trabajo. • Proporciona Librerías y Funciones que deberían hacer la vida del programador más feliz. • Ahorrar trabajo y tiempo. • Producir aplicaciones más fáciles de mantener. • Evitar código duplicado. • Crear Aplicaciones Multi-Capas
Wasp	<ul style="list-style-type: none"> • Esta versión incluye funciones para hacer la validación de formularios automático y fácil de hacer. • También incluye una biblioteca de funciones de validación que se pueden utilizar para validar las cosas comunes, como números de teléfono, códigos postales y fechas. • Estas funciones convenientemente emitir mensajes de error estándar que se pueden reemplazar o personalizar fácilmente.

<p style="text-align: center;">CodeIgniter</p>	<ul style="list-style-type: none"> • Sistema basado en Modelo -Vista - Controlador. • Extremadamente peso ligero. • Clases de base de datos con todas las funciones con soporte para varias plataformas. • Ajax. <input type="checkbox"/> Active Record. <input type="checkbox"/> Forma y validación de datos. • Seguridad y Filtrado XSS. • Gestión de la sesión. • Email Enviar clase. Apoya los accesorios, HTML / Texto email, múltiples protocolos (sendmail, SMTP y correo) y más.
<p style="text-align: center;">Seagull</p>	<ul style="list-style-type: none"> • Compatibilidad con PHP 4 y PHP 5. • ORM integrado. <input type="checkbox"/> Incorpora el patrón MVC. • Uso de templates. <input type="checkbox"/> Soporte para múltiples bases de datos. • Validación de datos. <input type="checkbox"/> Alto nivel de configuración. • Autenticación - integración de librerías PEAR. • PHP 4/5 compatibles. • Diseño por desempeño consciente. • Múltiples entradas y formatos de salida integrado.

BlueShoes	<ul style="list-style-type: none"> • Conectividad con bases de datos como MySQL, Oracle. • Desarrollo web con lenguaje php. • Gestión de contenidos. • Scripting PHP. □ Simplificar tareas comunes. • Reducción de las horas de programación.
Qcodo	<ul style="list-style-type: none"> • Modelo relacional objeto generación basada Código. • Basado en componentes visión orientada a eventos / controlador de la biblioteca. • Soporte AJAX basada en PHP totalmente integrada (sin JavaScripting necesario). • Orientado a objetos biblioteca de consulta de base de datos (no se requiere SQL). • Herramientas integradas de perfiles de base de datos. • Soporte de internacionalización.
Akelos	<ul style="list-style-type: none"> • Hace uso del modelo vista controlador (MVC). • Usa PHP4 Y PHP5 • Se ejecuta en casi todos los servidores de hosting.

PhpOpenbiz	<ul style="list-style-type: none"> • Hace uso de la modelo vista controlador (MVC). • Está orientado a las aplicaciones web para negocios
Zoop	<ul style="list-style-type: none"> • Hace uso de la modelo vista controlador (MVC). • Hace uso de AJAX. □ Es estable, escalable y portable□
Ash.MVC	<ul style="list-style-type: none"> • Hace uso de la modelo vista controlador (MVC). • establece un flujo de datos que da inicio en el navegador del cliente y culmina con éxito en el navegador web del cliente..
Diy	<ul style="list-style-type: none"> • Hace uso de la modelo vista controlador (MVC). • Tiene pocas restricciones y mucha flexibilidad. • Hace uso de componentes como plugins y AJAX. • se usa mayormente para el desarrollo de sitios web sociales basados en
Wact	<ul style="list-style-type: none"> • creación de aplicaciones web con una seguridad adecuada. • Hace uso de la modelo vista controlador (MVC).
Zend Framework	Código abierto, orientado a objetos, modelo vista controlador, gran rendimiento
Yii	Software libre, patrón de diseño modelo vista controlador, integración con JQuery
CakePHP	Compatible con PHP 4 y 5, validación integrada, Sintetización de datos
Laravel	Administrador de extensiones, completa documentación, contiene ORM
Symfony	Contiene una gran cantidad de plugins, integración con AJAX

Framework	PHP4	PHP5	MVC	Múltiple BD
KumbiaPHP	-	*	*	*
Wasp	-	*	*	-
CodeIgniter	*	*	*	*
Seagull	*	*	*	*
BlueShoes	*	*	*	*
Qcodo	-	*	*	*
Akelos	*	*	*	*
PhpOpenbiz	-	*	*	*
Zoop	*	*	*	*
Ash.MVC	-	*	*	-
Diy	-	*	*	-
Wact	*	*	*	*
Zend Framework	-	*	*	*
Yii	*	*	*	-
Laravel	-	*	*	*
CakePHP	*	*	*	*
Symfony	-	*	*	*

Framework	ORM	Plantillas	Cache	Ajax
KumbiaPHP	*	*	-	*
Wasp	-	*	-	*
CodeIgniter	-	*	*	-
Seagull	*	*	*	*
BlueShoes	-	*	-	-
Qcodo	-	*	-	*
Akelos	*	*	*	*
PhpOpenbiz	*	*	-	*
Zoop	-	*	*	*
Ash.MVC	-	*	-	-
Diy	*	*	*	*
Wact	-	*	-	-
Zend Framework	-	*	-	*
Yii	*	*	-	*
Laravel	*	*	*	*
CakePHP	-	*	*	*
Symfony	-	*	-	*

Imagen donde se muestra la tabla comparativa de los Framework por Luis Humberto, L. Y., y Alex Jhampier, R. H. (2017). ANÁLISIS COMPARATIVO DE FRAMEWORKS PHP PARA MEDIR EL RENDIMIENTO. Obtenido de Universidad Señor De Sipan website: <https://repositorio.uss.edu.pe/bitstream/handle/20.500.12802/4065/Llatas%20Yovera%2c%20Luis%20%26%20Rojas%20Herrera%2c%20Alex.pdf?sequence=11&isAllowed=y>.

FRAMEWORK	LARAVEL	CODEIGNITER	FUELPHP	CAKEPHP	SYMFONY	ZEND	PHALCON	YII	SLIM	PHPixie
COMPATIBILIDAD DE BASE DE DATOS	<ul style="list-style-type: none"> ✓ MySQL ✓ PostgreSQL ✓ SQLite ✓ SQL Server. 	<ul style="list-style-type: none"> ✓ MySQL ✓ PostgreSQL ✓ SQLite 	<ul style="list-style-type: none"> ✓ MySQL ✓ PostgreSQL ✓ SQLite ✓ MongoDB 	<ul style="list-style-type: none"> ✓ MySQL ✓ SQLite ✓ PostgreSQL ✓ SQL Server ✓ SQL Server 2008 ✓ Oracle 	<ul style="list-style-type: none"> ✓ MySQL ✓ PostgreSQL ✓ Microsoft SQL ✓ Microsoft SQL Server ✓ MongoDB 	<ul style="list-style-type: none"> ✓ MySQL ✓ PostgreSQL ✓ SQL Server ✓ SQLite. 	<ul style="list-style-type: none"> ✓ MySQL ✓ PostgreSQL ✓ SQLite ✓ MongoDB. 	<ul style="list-style-type: none"> ✓ MySQL ✓ MariaDB ✓ SQLite ✓ PostgreSQL ✓ CUBRID ✓ Oracle ✓ MSSQL 	<ul style="list-style-type: none"> ✓ MySQL 	<ul style="list-style-type: none"> ✓ MySQL ✓ PHPixie ✓ SQLite ✓ MongoDB.
Fuente:	("Laravel", p. https://laravel.com/docs/5.4/database)	("CodeIgniter", p. https://codeigniter.com/user_guide/database/configuration.html)	("FuelPHP", p. https://fuelphp.com/docs/#/api_docs)	("CakePHP", p. https://book.cakephp.org/3.0/en/orm/database-basics.html#database-configuration)	("Symfony", p. https://symfony.com/doc/current/doctrine.html)	("Zend Framework", n.d.-a, p. https://docs.zendframework.com/)	("PhalconPHP", p. https://phalconphp.com/es/)	("YiiFramework", p. http://www.yiiframework.com/doc-2.0/guide-db.html)	("Slim", p. https://www.slimframework.com/docs/tutorial/first.html)	("PHPixie", p. https://phpixie.com/components/database.html)

<p style="text-align: center;">SEGURIDAD</p>	<ul style="list-style-type: none"> ✓ Autenticación ✓ API Autenticación ✓ Autorización ✓ Encriptación ✓ Hashing ✓ Password Reset 	<ul style="list-style-type: none"> ✓ Seguridad URI. ✓ Filtrado XSS ✓ Protección CSRF ✓ Validación dato de entrada 	<ul style="list-style-type: none"> ✓ Filtrado de entrada ✓ Filtrado URI ✓ Filtrado XSS ✓ Codificación de Salida ✓ Protección de testigos CSRF ✓ Prevención de inyección SQL ✓ Un marco Auth Seguro 	<ul style="list-style-type: none"> ✓ Encriptación. ✓ Hashing de datos. ✓ Datos Aleatorios Seguros. ✓ CSRF ✓ Manejo de Devoluciones de llamada Blackhole ✓ Restricciones acciones SSL ✓ Restricciones comunicación entre controladores ✓ Prevención de manipulación de formularios 	<ul style="list-style-type: none"> ✓ Autenticación ✓ Autorización ✓ Encriptación de contraseñas ✓ Roles 	<ul style="list-style-type: none"> ✓ Autenticación ✓ Criptografía ✓ Control de acceso ✓ Serialización ✓ Contenedor de inyección SQL ✓ Validación. 	<ul style="list-style-type: none"> ✓ Hashing ✓ CSRF ✓ Generación Aleatoria de datos 	<ul style="list-style-type: none"> ✓ Autenticación ✓ Autorización ✓ Prevención de Cross-site Scripting ✓ Prevención de ataque de Cookies ✓ Prevención de Falsificación de Solitudes entre Sitios ✓ Inicio de sesión basado en Cookies ✓ Filtro de control de acceso ✓ Control de acceso de funciones 	<ul style="list-style-type: none"> ✓ Control de Inyección de dependencia 	<ul style="list-style-type: none"> ✓ Criptografía ✓ Tokens Aleatorio ✓ Entrega segura de cookies
---	---	---	---	---	---	---	--	--	---	---

								✓ Jerarquía de autorización		
Fuente:	("Laravel", p. https://laravel.com/docs/5.4/authentication)	("CodeIgniter," n.d., p. https://www.codeigniter.com/user_guide/general/security.html)	("FuelPHP," n.d., p. https://fuelphp.com/features)	("CakePHP", p. https://book.cakephp.org/3.0/en/security.html)	("Symfony", p. https://symfony.com/doc/current/security.html)	("Zend Framework", p. https://framework.zend.com/learn)	("PhalconPHP", p. https://docs.phalconphp.com/en/3.2/security)	("Yii", p. http://www.yiiframework.com/doc/guide/1.1/en/topics/auth)	("Slim", p. https://www.slimframework.com/)	("PHPixie", p. https://phpixie.com/)
VALIDACION INTEGRADA	<ul style="list-style-type: none"> ✓ RegisterController ✓ Auth ✓ Validación de Rutas 	✓ Validación datos de entrada	✓ Validación de formularios	✓ Validación de entrada	✓ Uso de Clase Validator	✓ Uso de Clase Validator	✓ Uso de clase Validation	<ul style="list-style-type: none"> ✓ Validación de datos de entrada ✓ Permite la creación de clases modelo para validación 	✓ Validación propia a cada desarrollador	✓ Validación simple de formularios

								✓ Validación de controlador		
Fuente:	("Laravel", p. https://laravel.com/docs/5.4/authentication#included-authentication)	("CodeIgniter", p. https://codeigniter.com/user_guide/general/security.html)	("FuelPHP", p. https://fuelphp.com/features)	("CakePHP", p. https://cakephp.org/#)	("Symfony", p. https://symfony.com/doc/current/validation.html)	("Zend Framework", p. https://framework.zend.com/learn)	("PhalconPHP", p. https://docs.phalconphp.com/en/3.2/db-models-validation)	("Yii", p. http://www.yiiframework.com/doc/guide/1.1/en/form-overview)	("Slim", p. https://www.slimframework.com/docs/)	("PHPixie", p. https://phpixie.com/components.html)
SISTEMA DE PLANTILLAS	<ul style="list-style-type: none"> ✓ Plantillas Blade ✓ Herencia de Plantillas ✓ Componentes y Ranuras ✓ Visualización de datos a través de las plantillas 	✓ Uso de Clase Parser para motor de plantillas.	✓ Uso de Paquete Parser	✓ Uso de plantillas XML	✓ Sistema de Plantillas Twig.	✓ Uso de plantillas XML	✓ Uso de plantillas XML	<ul style="list-style-type: none"> ✓ CWebService ✓ CWebServiceAction ✓ WSDL 	✓ Uso de plantilla Twig-View	✓ Definidas por el mismo desarrollador
RAPIDO Y FLEXIBLES	<ul style="list-style-type: none"> ✓ Estructuras de control 									

	<ul style="list-style-type: none"> ✓ Incluye Sub – Vistas ✓ Láminas de Extensión 									
Fuente:	("Laravel", p. https://laravel.com/docs/5.4/blade)	("CodeIgniter", p. https://codeigniter.com/user_guide/libraries/parser.html)	("FuelPHP", p. https://fuelphp.com/features)	("CakePHP", p. https://book.cakephp.org/3.0/en/index.html)	("Symfony", p. https://symfony.com/doc/current/templating.html)	("Zend Framework", p. https://framework.zend.com/learn)	("PhalconPHP", p. https://docs.phalconphp.com/en/3.2/db-layer)	("YiiFramework", p. http://www.yiiframework.com/doc/guide/1.1/en/topic-services)	("Slim", p. https://www.slimframework.com/docs/features/templates.html)	("PHPixie", p. https://phpixie.com/)
COMPATIBILIDAD PARA PHP	PHP >= 5.6.4	PHP >= 5.6	PHP = 5.3	PHP >= 5.6	PHP = 5.4	PHP >= 5.6	PHP >= 5.5	PHP >= 5.4	PHP 5.5	PHP = 5.3
Fuente:	("Laravel", p. https://laravel.com/docs/5.4)	("CodeIgniter", p. https://codeigniter.com/user_guide/general/requirements.html)	("FuelPHP", n.d., p. https://fuelphp.com/features)	("CakePHP", p. https://book.cakephp.org/3)	("Symfony", p. https://symfony.com/doc/current/)	("Zend Framework", p. https://docs.zend.com/)	("PhalconPHP", p. https://docs.phalconphp.com/en/3.2/installation)	("YiiFramework", p. http://www.yiiframework.com/doc/start/installation)	("Slim", p. https://www.slimframework.com/docs/start/installation)	("PHPixie", p. https://phpixie.com/docs/php-environment)

				.0/en/index.html)	setup.html)	s.zendframework.com/tutorials/getting-started/overview/)		om/doc-2.0/guide-start-installation.html)	allation.html)	nt-setup.html)
DOCUMENTACION	<ul style="list-style-type: none"> ✓ Detallada ✓ Entendible ✓ Forums ✓ Twiter ✓ Congresos Online 	<ul style="list-style-type: none"> ✓ Detallada ✓ Entendible 	<ul style="list-style-type: none"> ✓ Muy General ✓ Poco Detallada ✓ Foro. 	<ul style="list-style-type: none"> ✓ Detallada ✓ Entendible ✓ Comunidad 	<ul style="list-style-type: none"> ✓ Muy General ✓ Poco detallada ✓ Comunidad 	<ul style="list-style-type: none"> ✓ Muy General ✓ Blog 	<ul style="list-style-type: none"> ✓ Muy General ✓ Blog 	<ul style="list-style-type: none"> ✓ Detallada ✓ Libros ✓ Tutoriales ✓ Forums ✓ Chat ✓ Wiki ✓ Contribución de otros desarrolladores para la mejora del framework 	<ul style="list-style-type: none"> ✓ Muy General ✓ Blog 	<ul style="list-style-type: none"> ✓ Muy General ✓ Poco Detallada ✓ Blog
Fuente:	("Laravel," n.d., p. https://lara	("CodeIgniter", p. https://codeignit	("FuelPHP", p. https://fue	("Cake PHP", p. https://s	("Symfony", p. https://s	("Zend Framework", p. https://docs	("PhalconPHP", p. https://docs	("YiiFramework", p. http://ww	("Slim", p. https://www.slimfr	("PHPixie", p. https://

	vel.com/docs/5.4)	er.com/user_guide/)	lphp.com/docs/)	https://book.cakephp.org/3.0/en/index.html)	ymfony.com/docs/current/index.html)	ework," n.d.-a, p. https://framework.zend.com/learn)	phalconphp.com/en/3.2)	w.yiiframework.com/doc/)	amework.com/docs/)	phpixie.com/docs.html)
RUTEO DE URL FLEXIBLE	<ul style="list-style-type: none"> ✓ Generación de URLs a rutas asignadas ✓ Grupos de rutas ✓ Enrutamiento de subdominio ✓ Enlaces del modelo de ruta ✓ Acceso a la ruta actual 	<ul style="list-style-type: none"> ✓ Reglas de enrutamiento estático ✓ Rutas Reservadas 	<ul style="list-style-type: none"> ✓ Hace uso de la clase Route 	<ul style="list-style-type: none"> ✓ Hace uso de la clase Routing 	<ul style="list-style-type: none"> ✓ Hace uso de la clase Routing 	<ul style="list-style-type: none"> ✓ Hace uso de la clase Routing 	<ul style="list-style-type: none"> ✓ Hace uso de la clase Routing 	<ul style="list-style-type: none"> ✓ Creación de URLs dinámica ✓ Permite definir reglas para URL ✓ Permite Parametrizar Rutas ✓ Permite agregar sufijos a URL ✓ Permite asignar Reglas dinámicamente ✓ Normalización de URL 	<ul style="list-style-type: none"> ✓ Enrutamiento HTTP 	<ul style="list-style-type: none"> ✓ Enrutamiento HTTP

Fuente:	("Laravel", p. https://laravel.com/docs/5.4/routing)	("CodeIgniter", p. https://codeigniter.com/user_guide/general/routing.html)	("FuelPHP", p. https://fuelphp.com/features)	("CakePHP", p. https://api.cakephp.org/3.4/namespaces-Cake.Routing.html)	("Symfony", p. https://symfony.com/doc/current/routing.html)	("Zend Framework", p. https://symfony.com/doc/current/routing.html)	("PhalconPHP", p. https://docs.phalconphp.com/en/3.2/routing)	("YiiFramework", p. http://www.yiiframework.com/doc-2.0/guide-runtime-routing.html)	("Slim", p. https://www.slimframework.com/)	("PHPixie", n.d.-b, p. https://phpixie.com/components/httpl)
SCAFFOLDING	<ul style="list-style-type: none"> ✓ Scaffolding JavaScript ✓ Scaffolding CSS 	✓ No implementa	✓ Scaffolding Básico	✓ No implementa	✓ No implementa	✓ No implementa	✓ No implementa	<ul style="list-style-type: none"> ✓ Generación de Class Active Record ✓ Generación de CRUD ✓ Generador de Modulo ✓ Generador de Controller ✓ Generador de 	✓ No implementa	✓ No implementa

								Extensi ón		
Fuente:	("Laravel", p. https://laravel.com/docs/5.4/frontend)	("CodeIgniter", p. https://codeigniter.com/user_guide/)	("FuelPHP," n.d., p. https://fuelphp.com/features)	("CakePHP", p. https://book.cakephp.org/3.0/en/index.html)	("Symfony", p. https://symfony.com/doc/current/index.html)	("Zend Framework", p. https://framework.zend.com/learn)	("PhalconPHP", p. https://docs.phalconphp.com/en/3.2)	("YiiFramework", p. http://www.yiiframework.com/doc-2.0/guide-start.html)	("Slim", p. https://www.slimframework.com/docs/)	("PHPixie", p. https://phpixie.com/docs.html)
MAPEO DE OBJETOS RELACIONAL - ORM	✓ Si implementa (Eloquent, Relationships, Collections, Mutators, Serialization)	✓ No implementa	✓ No implementa	✓ Si implementa	✓ Si implementa	✓ No implementa	✓ Si implementa	✓ Si implementa (Active Record)	✓ No implementa	✓ No implementa
Fuente:	("Laravel", p. https://laravel.com/docs/5.4/eloquent)	("CodeIgniter", p. https://codeigniter.com/user_guide/)	("FuelPHP", p. https://fuelphp.com/docs/)	("CakePHP", p. https://book.cakephp.org/3.0/en/orm.html)	("Symfony", p. https://symfony.com/doc/current/doctrine.html)	("Zend Framework", p. https://framework.zend.com/manual/2.3/en/user-guide/database-and-	("PhalconPHP", p. https://docs.phalconphp.com/en/3.2)	("YiiFramework", p. http://www.yiiframework.com/doc-2.0/guide-start.html)	("Slim", p. https://www.slimframework.com/)	("PHPixie", p. https://phpixie.com/docs.html)

						models.html)				
FACIL LECTURA DE CODIGO	✓ Entendible	✓ Entendible	✓ Difícil de Comprender	✓ Difícil de Comprender	✓ Difícil de Comprender	✓ Difícil de Comprender	✓ Entendible	✓ Entendible	✓ Difícil de Comprender	✓ Difícil de Comprender
Fuente:	("Laravel," n.d., p. https://laravel.com/docs/5.4/views#creating-views)	("CodeIgniter", p. https://codeigniter.com/user_guide/general/views.html)	("FuelPHP", p. https://fuelphp.com/docs/general/packages.html)	("CakePHP", p. https://book.cakephp.org/3.0/en/tutorials-and-examples/bookmarks/intro.html)	("Symfony", p. https://symfony.com/doc/current/templating.html)	("Zend Framework", p. https://framework.zend.com/learn)	("PhalconPHP", p. https://docs.phalconphp.com/en/3.2/views)	("YiiFramework," n.d., p. http://www.yiiframework.com/doc-2.0/guide-structure-views.html#layouts)	("Slim," n.d.-b, p. https://www.slimframework.com/docs/)	("PHPixie", p. https://phpixie.com/components/template.html)

Imagen de tabla comparativa de los Framework relacionados con Php, Francia Peralta, N. F. (2019). Análisis comparativo de frameworks open source para el nivel de productividad en los proyectos de desarrollo de software web. Obtenido de Universidad Tecnológica del Perú website: https://repositorio.utp.edu.pe/bitstream/handle/20.500.12867/2736/Norbil%20Francia_Trabajo%20de%20Investigacion_Bachiller_2019.pdf?sequence=1&isAllowed=y.

FRAMEWORK	LARAVEL	CODEIGNITER	FUELPHP	CAKEPHP	SYMFONY	ZEND	PHALCON	YII	SLIM	PHPixie
VALIDACION INTEGRADA	<ul style="list-style-type: none"> ✓ Register controller ✓ Auth ✓ Validación de Rutas ✓ Validación de datos de entrada 	<ul style="list-style-type: none"> ✓ Validación de datos de entradas 	<ul style="list-style-type: none"> ✓ Validación de formularios 	<ul style="list-style-type: none"> ✓ Validación de entrada 	<ul style="list-style-type: none"> ✓ Uso de Clase Validator 	<ul style="list-style-type: none"> ✓ Uso de Clase Validator 	<ul style="list-style-type: none"> ✓ Uso de clase Validation 	<ul style="list-style-type: none"> ✓ Validación de datos de entrada ✓ Permite la creación de clases modelo para validación ✓ Validación de controlador 	<ul style="list-style-type: none"> ✓ Validación propia a cada desarrollador 	<ul style="list-style-type: none"> ✓ Validación simple de formularios
Fuente:	("Laravel", p. https://laravel.com/docs/5.4/authentication#included-authenticating)	("CodeIgniter", p. https://codeigniter.com/user_guide/general/security.html)	("FuelPHP", p. https://fuelphp.com/features)	("CakePHP", p. https://cakephp.org/#)	("Symfony", p. https://symfony.com/doc/current/validation.html)	("Zend Framework", p. https://framework.zend.com/learn)	("PhalconPHP", p. https://docs.phalconphp.com/en/3.2/dbmodels/validation)	("Yii", p. http://www.yiiframework.com/doc/guide/1.1/en/form-overview)	("Slim", p. https://www.slimframework.com/docs/)	PHPixie", p. https://phpixie.com/components.html)

SISTEMA DE PLANTILLAS RAPIDO Y FLEXIBLES	<ul style="list-style-type: none"> ✓ Plantillas Blade ✓ Herencia de Plantillas ✓ Componentes y Ranuras ✓ Visualización de datos a través de las plantillas ✓ Estructuras de control 	<ul style="list-style-type: none"> ✓ Uso de Clase Parser para motor de plantillas. 	<ul style="list-style-type: none"> ✓ Uso de Paquete Parser 	<ul style="list-style-type: none"> ✓ Uso de plantillas XML 	<ul style="list-style-type: none"> ✓ Sistema de Plantillas Twig. 	<ul style="list-style-type: none"> ✓ Uso de plantillas XML 	<ul style="list-style-type: none"> ✓ Uso de plantillas XML 	<ul style="list-style-type: none"> ✓ CWebService ✓ CWebServiceAction ✓ WSDL 	<ul style="list-style-type: none"> ✓ Uso de plantilla Twig-View 	<ul style="list-style-type: none"> ✓ Definidas por el mismo desarrollador
	<ul style="list-style-type: none"> ✓ Incluye Sub-Vistas Láminas de Extensión 									

Fuente:	("Laravel", p. https://laravel.com/docs/5.4/blade)	("CodeIgniter", p. https://codeigniter.com/user_guide/libraries/parser.html)	("FuelPHP", p. https://fuelphp.com/features)	("Cake PHP", p. https://book.cakephp.org/3.0/en/index.html)	("Symfony", p. https://symfony.com/doc/current/templating.html)	("Zend Framework", p. https://framework.zend.com/learn)	("PhalconPHP", p. https://docs.phalconphp.com/en/3.2/dblayer)	("YiiFramework", p. http://www.yiiframework.com/doc/guide/1.1/en/topics/services)	("Slim", p. https://www.slimframework.com/docs/features/templates.html)	("PHPixie", p. https://phpixie.com/)
----------------	--	--	--	---	--	---	---	---	---	--

DOCUMENTACION	<ul style="list-style-type: none"> ✓ Detallada ✓ Entendible ✓ Forums ✓ Twiter ✓ Congresos Online ✓ Contribución de otros desarrolladores para la mejora del framework 	<ul style="list-style-type: none"> ✓ Detallada ✓ Entendible 	<ul style="list-style-type: none"> Muy General Poco Detallada ✓ Foro. 	<ul style="list-style-type: none"> ✓ Detallada ✓ Entendible ✓ Comunidad 	<ul style="list-style-type: none"> ✓ Muy General ✓ Poco detallada ✓ Comunidad 	<ul style="list-style-type: none"> ✓ Muy General ✓ Blog 	<ul style="list-style-type: none"> ✓ Muy General ✓ Blog 	<ul style="list-style-type: none"> ✓ Detallada ✓ Libros ✓ Tutoriales ✓ Forums ✓ Chat ✓ Wiki ✓ Contribución de otros desarrolladores para la mejora del framework 	<ul style="list-style-type: none"> Muy General ✓ Blog 	<ul style="list-style-type: none"> ✓ Muy General ✓ Poco Detallada ✓ Blog
----------------------	---	---	--	--	--	---	---	---	---	---

Fuente:	("Laravel," n.d., p. https://laravel.com/docs/5.4)	("CodeIgniter", p. https://codeigniter.com/user_guide/)	("FuelPHP", p. https://fuelphp.com/docs/)	("Cake PHP", p. https://book.cakephp.org/3.0/en/index.html)	("Symfony", p. https://symfony.com/doc/current/index.html)	("Zend Framework," n.d.a, p. https://framework.zend.com/learn)	("PhalconPHP", p. https://docs.phalconphp.com/en/3.2)	("YiiFramework", p. http://www.yiiframework.com/doc/)	("Slim", p. https://www.slimframework.com/docs/)	("PHPIXIE", p. https://phpixie.com/docs.html)
----------------	--	--	--	---	--	--	---	---	---	--

SCAFFOLDING	<ul style="list-style-type: none"> ✓ Scaffolding ✓ JavaScript ✓ Scaffolding CSS ✓ Generación de ✓ CRUD 	✓ No implementa	✓ Scaffolding Básico	✓ No implementa	✓ No implementa	✓ No implementa	✓ No implementa	<ul style="list-style-type: none"> ✓ Generación de Class Active Record ✓ Generación de CRUD ✓ Generador de Modulo ✓ Generador de Controler ✓ Generador de Extensión 	✓ No implementa	✓ No implementa
--------------------	---	-----------------	----------------------	-----------------	-----------------	-----------------	-----------------	--	-----------------	-----------------







Fuente:	("Laravel", p. https://laravel.com/docs/5.4/frontend)	("CodeIgniter", p. https://codeigniter.com/user_guide/)	("FuelPHP", n.d., p. https://fuelphp.com/features)	("CakePHP", p. https://book.cakephp.org/3.0/en/index.html)	("Symfony", p. https://symfony.com/doc/current/index.html)	("Zend Framework", p. https://framework.zend.com/learn)	("PhalconPHP", p. https://docs.phalconphp.com/en/3.2)	("YiiFramework", p. http://www.yiiframework.com/doc2.0/guidestart.html)	("Slim", p. https://www.slimframework.com/docs/)	("PHPIXIE", p. https://phpixie.com/docs.html)
MAPEO DE OBJETOS RELACIONAL - ORM	✓ Si implementa (Eloquent, Relationships, Collections, Mutators, Serialization)	✓ No implementa	✓ No implementa	✓ Si implementa	✓ Si implementa	✓ No implementa	✓ Si implementa	✓ Si implementa (Active Record)	✓ No implementa	✓ No implementa
Fuente:	("Laravel", p. https://laravel.com/docs/5.4/eloquent)	("CodeIgniter", p. https://codeigniter.com/user_guide/)	("FuelPHP", p. https://fuelphp.com/docs/)	("CakePHP", p. https://book.cakephp.org/3.0/en/orm.html)	("Symfony", p. https://symfony.com/doc/current/doctrine.html)	("Zend Framework", p. https://framework.zend.com/manual/2.3/en/userguide/database-and-models.html)	("PhalconPHP", p. https://docs.phalconphp.com/en/3.2)	("YiiFramework", p. http://www.yiiframework.com/doc2.0/guide-index.html)	("Slim", p. https://www.slimframework.com/)	("PHPIXIE", p. https://phpixie.com/docs.html)

FACIL LECTURA DE CODIGO	✓ Entendible	✓ Entendible	✓ Difícil de Comprender	✓ Difícil de Comprender	✓ Difícil de Comprender	✓ Difícil de Comprender	✓ Entendible	✓ Entendible	✓ Difícil de Comprender	✓ Difícil de Comprender
Fuente:	("Laravel," n.d., p. https://laravel.com/docs/5.4/views#creating-views)	("CodeIgniter", p. https://codeigniter.com/user_guide/general/views.html)	("FuelPHP", p. https://fuelphp.com/docs/general/packages.html)	("CakePHP", p. https://book.cakephp.org/3.0/en/tutorialsandexamples/bookmarks/intro.html)	("Symfony", p. https://symfony.com/doc/current/templating.html)	("Zend Framework", p. https://framework.zend.com/learn)	("PhalconPHP", p. https://docs.phalconphp.com/en/3.2/views)	("YiiFramework", n.d., p. http://www.yiiframework.com/doc2.0/guide-structure-views.html#layouts)	("Slim", n.d.-b, p. https://www.slimframework.com/docs/)	("PHPixie", p. https://phpixie.com/components/template.html)

Anexo 11. Imagen de tabla comparativa de los gestores de bases de datos por, Roberto, I. (2019). Análisis, diseño e implementación de un sistema para la gestión de postulantes de la carrera de Computación de la Universidad Católica de Santiago de Guayaquil. 223.180. <https://doi.org/http://repositorio.ucsg.edu.ec/handle/3317/13864>.

	Sqlite	Mysql	Postgresql
Arquitectura	Basado en archivos (incrustado)	Servidor de cliente	Servidor de cliente
Servidor OS	Ninguno (sin servidor)	FreeBSD, Linux, OS X, Solaris, Windows	FreeBSD, HP-UX, Linux, NetBSD, OpenBSD, OS X, Solaris, Unix, Windows
Lenguaje de programación	C	C, C ++	C
Lenguajes de programación soportados	ActionScript, Ada, Basic, C, C #, C ++, D, Delphi, Forth, Fortran, Haskell, Java, JavaScript, Lisp, Lua, MATLAB, PHP, PL / SQL	C, C ++, Delphi, Perl, Java, Lua, .NET, Node.js	NET, C, C ++, Delphi, Java, Perl, PHP, Python, Tcl
DBaaS	Ninguna	Base de datos de Azure para MySQL, Amazon RDS para MySQL, Google Cloud SQL para MySQL	Base de datos de Azure para PostgreSQL, Amazon RDS para PostgreSQL, Google Cloud SQL para PostgreSQL
Cientes clave	Adobe, Facebook y Apple	GitHub, Facebook y YouTube	Cloudera, Instagram y ViaSat

Imágenes de Tablas comparativa de los Gestores de base de datos por, Jenner, S. (2020). Análisis comparativo de sistemas gestores de bases de datos postgresql y mysql en procesos crud. Uss.edu.pe. <https://doi.org/https://hdl.handle.net/20.500.12802/7012>.

Rank			DBMS	Database Model	Score		
Apr 2019	Mar 2019	Apr 2018			Apr 2019	Mar 2019	Apr 2018
1.	1.	1.	Oracle	Relational, Multi-model	1279.94	+0.80	-9.85
2.	2.	2.	MySQL	Relational, Multi-model	1215.14	+16.89	-11.26
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	1059.96	+12.11	-35.55
4.	4.	4.	PostgreSQL	Relational, Multi-model	478.72	+8.91	+83.25
5.	5.	5.	MongoDB	Document	401.98	+0.64	+60.57
6.	6.	6.	IBM Db2	Relational, Multi-model	176.05	-1.15	-12.89
7.	 8.	 9.	Redis	Key-value, Multi-model	146.38	+0.25	+16.27
8.	 9.	8.	Elasticsearch	Search engine, Multi-model	146.00	+3.21	+14.64
9.	 7.	 7.	Microsoft Access	Relational	144.65	-1.55	+12.43
10.	10.	 11.	SQLite	Relational	124.21	-0.66	+8.23

PLAN DE DESARROLLO DE SOFTWARE V1.0

Propuesta de un sistema de gestión de inventarios para la farmacia del Hospital del Día Nova Salud de la ciudad de Tulcán.

Generado por

Darwin Fernando Tipan Viveros

GENERALIDADES DEL PROYECTO

Descripción del proyecto

El proyecto denominado “Propuesta de un sistema de gestión de inventarios para la farmacia del Hospital del Día Nova Salud de la ciudad de Tulcán” está orientado a desarrollar un sistema que permita el control de los materiales y accesorios del departamento de farmacia, basado en arquitectura web modelo vista controlador, en este documento se detalla la información general del proyecto, su organización y gestión.

Propósito

El propósito de este documento es brindar un panorama general de la organización y gestión del proyecto, además sirve como complemento a la documentación generada por la metodología utilizada para el desarrollo de software XP (Programación Extrema).

Historial de revisiones

Tabla 31.

Historial de revisiones

Fecha	Versión	Descripción	Autor	Aprobado por
06/09/2021	1.0	Revisión y aprobación del proyecto	Darwin Tipan	Lic.Carmen Zuñiga

Alcance

Desarrollo de una propuesta tecnológica con herramientas de programación flexibles y de fácil interacción para el usuario, este documento muestra la gestión del equipo de investigación durante la codificación del sistema hasta la finalización del software.

Objetivos

- Recopilar los requerimientos funcionales utilizando las herramientas de la metodología XP.
- Seleccionar las herramientas de programación adecuadas para el desarrollo del sistema de control de inventario.
- Adoptar los estándares que dicta la metodología de desarrollo de software para la ejecución del proyecto.
- Desarrollar un sistema de gestión de control de inventario que integre los módulos requeridos por el departamento de Farmacia.

Suposiciones y restricciones

En base a la entrevista no estructurada se recopiló las siguientes restricciones:

1. Se debe analizar a fondo el funcionamiento de los procesos administrativos y toda la información que esté involucrada.
2. El sistema de control de inventario debe ser una plataforma web que integre los módulos que el departamento de Farmacia solicite.
3. La interfaz de usuario debe ser fácil de utilizar para el usuario administrador.

Evolución del plan de desarrollo de software

El plan de desarrollo de software se revisará cada vez que haya una modificación de gran magnitud en el sistema de control de inventario.

ORGANIZACIÓN DEL PROYECTO

Recursos humanos y profesionales

Tabla 32.

Recursos humanos y profesionales

Quien	Información de contacto	Celular
Darwin Fernando Tipan Viveros	darwin958095@hotmail.com	0998450851

Roles y responsabilidades

Tabla 33.

Roles y responsabilidades

Rol	Responsabilidad	Asumido por
Consultor	Ayuda al equipo de trabajo a solventar problemas en el área de la informática	Msc. Samuel Lascano
Programadores	Desarrollan código funcional que solventa los requerimientos del cliente y se aseguran que el software sea seguro y confiable	Darwin Tipan
Cliente	Define los requerimientos del software y facilita la información necesaria para el desarrollo.	Lic.Carmen Zuñiga

GESTIÓN DEL PROYECTO

Plan del proyecto

Tabla 34.

Plan de proyecto

Fase	Iteraciones	Duración
Fase de diseño	1	1 mes
Fase de diseño	1	4 meses
Fase de codificación	1	4 meses
Fase de pruebas	1	1 mes

Fases y líneas base

Tabla 35.

Fases y líneas base

Fase	Objetivo asociado	Riesgos contemplados
Fase de planificación	En primera fase se recopiló información acerca de los procesos administrativos y las herramientas a utilizar en el desarrollo del sistema	Recolectar información de los requerimientos del cliente para planificar actividades.
Fase de diseño	En fase de diseño se analizan los componentes que serán utilizados en la distribución del software, estos permitirán que se optimice la estructura de forma constante.	Diseñar prototipos de diseño con base en los requerimientos obtenidos en la entrevista no estructurada.
Fase de codificación	Se realizará la programación en lo que permitirá que exista una retroalimentación mutua e inmediata sobre el desarrollo del software, además se identificará de forma más efectiva los errores cometidos para su solución.	Desarrollar las funcionalidades de código para integrar la información de los procesos administrativos del departamento.
Fase de pruebas	Las pruebas unitarias y de aceptación, permitirán crear un ciclo de desarrollo más rápido, esto permitirá que la aprobación del cliente sea validada por completo.	Realizar pruebas unitarias de las funcionalidades más importantes del sistema y pruebas de aceptación para garantizar la satisfacción del cliente.

Objetivos y riesgos de cada fase

Tabla 36.

Objetivos y riesgos

Fase	Objetivo asociado	Riesgos contemplados
Fase de planificación	Recolectar información de los requerimientos del cliente	Falta de compromiso por parte del cliente, La información obtenida no concuerda con los requerimientos, Cambio de requerimientos por parte del cliente
Fase de diseño	Diseñar prototipos de diseño con base en los requerimientos.	Fallo en el diseño de interfaces, el diseño no satisface las expectativas del cliente.
Fase de codificación	Desarrollar las funcionalidades de código para integrar la información de los procesos administrativos del departamento.	Pérdida del código fuente, Poca experiencia de los desarrolladores, inconvenientes con la base de datos.
Fase de pruebas	Realizar pruebas unitarias de las funcionalidades más importantes del sistema y pruebas de aceptación para garantizar la satisfacción del cliente.	Obtener falsos positivos en pruebas unitarias, Falta de colaboración del cliente para realizar las pruebas de aceptación.

Diagrama de Gantt

Tabla 37.

Diagrama de Gantt

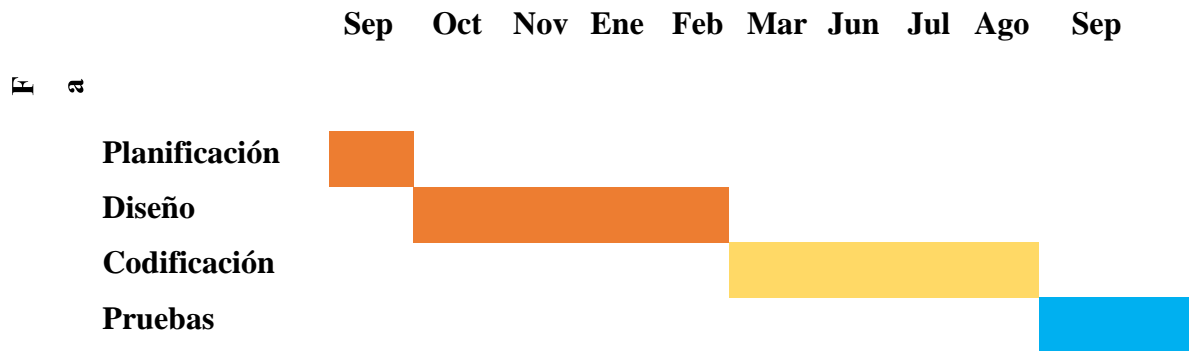
Nombre de la fase	Fecha de inicio	Duración en meses	Fecha fin
Planificación	02/09/2019	1	29/09/2019
Diseño	18/10/2020	4	29/01/2021
Codificación	01/02/2021	4	22/06/2021
Pruebas	01/08/2021	1	06/09/2021

Calendario del proyecto

Tabla 38.

Calendario de proyecto

Meses



Control de desviación a la planificación

Tabla 39.

Control de desviación a la planificación

Item	Probabilidad	Impacto
1	Improbable	Bajo
2	Poco Probable	Mínimo
3	Probabilidad media	Medio
4	Casi probable	Alto
5	Probable	Muy alto

Metodología Xp

Fase de planificación

Esta fase incluye la definición de los roles que juegan los individuos en el proyecto utilizando modelos XP, así como las equivalencias de tiempo que se utilizaron en las herramientas metodológicas. Adicionalmente, se elaboró un plan de entrega en base al tiempo que tomó desarrollar el proyecto, dando como resultado un total de 7 meses, el cual se dividió de la misma manera.

Estudio de factibilidad

Factibilidad organizacional

Aspectos generales de la organización

Institución: Hospital del día Nova Salud

Ubicación geográfica: Tulcán-Carchi

Área: Departamento de Farmacia

Sistema: Sistema de control de inventario

Objeto social: Servicio privado

Recursos del software

Tabla 42.

Recursos de software

Tip de recurso	Nombre del recurso	Descripción	Cantidad
Software	MySQL	Base de datos	1
	PhpMyAdmin	Gestor de base de datos	1
	Bootstrap	Framework de Estilo	1
	PHP CodeIgniter V4	Lenguaje de programación	1
	Vs Code	Entorno de programación	1
	Git y GitHub	Sistema de control de versiones	1
	Microsoft office	Herramienta ofimática	1

El investigador dispone de todas las herramientas necesarias para el desarrollo del proyecto, así como de una conexión a internet, lo que facilita la comunicación dentro del equipo. Como resultado, el investigador concluye que los recursos son suficientes y que existe factibilidad técnica.

Tabla 43.
Recursos de hardware

Tip de recurso	Nombre del recurso	Descripción	Cantidad
Hardware	Equipo de computación	Laptop hp da1500la Disco duro de 1TB HDD Memoria Ram 16 GB	1
	Servidor	Servidor HP xampp serie v.3.2.2	1

Factibilidad Económica.

En el presupuesto del proyecto se tomó en cuenta recursos de hardware, software, talento humano y materiales de oficina.

Tabla 44.
Costos de equipos

Descripción	Cantidad	Costo real costos de hardware	Costo referencial
Equipo de computación	1	00	2500
Servidor de aplicación (host)	1	00	25.99 alojamiento
Total, del costo	2	00	2525.99

Tabla 45.
Recursos de oficina

Total, de talento humano	Cantidad	Costo real
Internet	1	29.99
Programadores	1	00
Útiles de oficina	1	360
Varios	5	58.99
Total, de costos	8	448.98

Factibilidad Operativa.

Situación actual

El control y los registros se mantienen en hojas de cálculo en el departamento de farmacia, lo que dificulta el acceso y la recuperación. Esto provoca un retraso en el control de los materiales registrados y, en caso de falla, la información puede perderse por falta de métodos de almacenamiento adecuados.

Los procesos asociados con el módulo de control consumen mucho tiempo para el personal de farmacia ya que requieren abrir muchas carpetas para ver el historial de registro y los registros deben actualizarse manualmente cada vez que se recibe un nuevo material.

Situación ideal

El sistema de control de inventario para la gestión de los materiales y accesorios del departamento de farmacia va a integrar los módulos de control de inventario, venta y abastecimiento. Lo que permite centralizar la información de los registros y facilitar la búsqueda de datos e historiales. Por parte de la administración se cuenta con el apoyo necesario y se va a trabajar de forma coordinada con la administradora del hospital para obtener los requerimientos necesarios que cumplan con las funcionalidades específicas del sistema.

Roles

Tabla 46.
Roles del proyecto

Nombre	Descripción	Rol XP
Msc. Samuel Lascano	Docente Tutor	Consultor
Darwin Tipan	Investigador	Programador
Lic. Carmen Zúñiga	Administradora	Cliente
Enriques Telpiz	Bodeguero	Cliente

Estimación de tiempo

Tabla 47.

Estimación de tiempo

Estimación	Días	Horas
Primer mes	23	12
Segundo mes	23	12
Tercer mes	23	12
Cuarto mes	23	12
Quinto mes	23	12
Sexto mes	23	24
Séptimo mes	23	24

Módulos del sistema

Notificaciones

Venta

- a. Búsqueda por nombre o código de barra

Ventas

- a. Ventas
- b. Ventas crédito
- c. Por entregar
- d. Por cobrar
- e. Ventas canceladas
- f. Cotizaciones

Compras

- a. Nueva compra
- b. Compras
- c. Por recibir
- d. Por pagar
- e. Compras canceladas

Catálogos

- a. Productos
- b. Categorías
- c. Marcas
- d. Clientes

- e. Proveedores
- f. Sucursales
- g. Administrar precios “sucursales”

Herramientas

- a. Contactos
- b. Mensajes “sucursales”

Finanzas básicas

- a. Crédito
- b. Balance
- c. Gastos
- d. Caja

Módulo principal Inventario

- a. Inventario principal
- b. Abastecer
- c. Valor del inventario
- d. Buscar productos
- e. Inventario global
- f. Traspasar
- g. Devolución
- h. Traspasos
- i. Devoluciones

Reportes

- a. Inventario
- b. Por categorías
- c. Ventas
- d. Compras
- e. Reporte de pagos “crédito”
- f. Clientes populares
- g. Vendedores populares
- h. Productos populares

Administración

- a. Usuarios
- b. Configuraciones
- c. Importar Datos

Historias de usuario.

Módulo N.º 1. Administración de usuarios

Tabla 48.

Historia de usuario 1

HISTORIA DE USUARIO	
Número: 1	Usuario: Administrador
Nombre historia: Acceso al sistema de administración	
Prioridad: Alta	Riesgo: Medio
Estimación: 1	Iteración: 1
Responsable: Darwin Tipan	
Descripción: La administradora tendrá facilidad de acceso a los módulos completos del sistema con sus credenciales de usuario	
Detalle: La ventana debe tener las siguientes características	
<ul style="list-style-type: none">- Dos campos de tipo texto para ingresar con usuario y contraseña- Un botón de “Ingresar” que redirigirá al panel principal	

Tabla 49.

Historia de usuario 2

HISTORIA DE USUARIO	
Número: 2	Usuario: Administrador
Nombre historia: Panel de control	
Prioridad: Alta	Riesgo: Medio
Estimación: 2	Iteración: 1
Responsable: Darwin Tipan	
Descripción: La administradora podrá observar y realizar acciones sobre los módulos correspondientes.	
Detalle: El panel de control debe tener las siguientes características.	
<ul style="list-style-type: none">- Una barra estadística, barra de botones de acceso rápido.- Campos de acción, Inició, Notificaciones, vender, ventas(ventas, ventas crédito, Por entregar, Por cobrar, Ventas Canceladas, Cotizaciones), Compras (Nueva*, Compras, Por Recibir, Por Pagar, Compras	

Canceladas), Catálogos (Productos, categorías, Marcas, Clientes, Proveedores, Sucursales, Administrar Precios) Herramientas(Contactos, Mensajes), Finanzas (Crédito, Balance, Gastos, Caja), Inventario (Inventario Principal, Abastecer, Valor del Inventario, Buscar Productos, Inventario Global, Traspasar, Devoluciones, Traspasos, Devoluciones) Reportes (Log de inventario, Por Categorías, Ventas, Compras, Reporte de pagos “Crédito”, Clientes Populares, Vendedores Populares, Productos Populares, Administración (Usuarios, Configuración, Importar datos))

Tabla 50.
Historia de usuario 3

HISTORIA DE USUARIO	
Número: 3	Usuario: Administrador
Nombre historia: vender	
Prioridad: Alta	Riesgo: Alto
Estimación: 3	Iteración: 1
Responsable: Darwin Tipan	
Descripción: La administradora debe poder realizar la venta dentro del sistema.	
Detalle: La acción de vender debe tener las siguientes características:	
<ul style="list-style-type: none"> - Dos campos para realizar la búsqueda - Un botón con el nombre de “buscar” que permita realizar la orden - Una tabla donde se muestren los resultados, dentro de esta se deberá ubicar un botón para agregar la cantidad deseada. - Campos de texto para llenar los siguientes datos, “No. Factura, almacén, cliente, Pago, Entrega, Forma de pago, Descuento, Efectivo.” - Campos que se detalle los totales. - Dos botones con el texto “Cancelar, Finalizar venta” que permitan realizar la orden. - Mostrará la venta realizada. 	

Tabla 51.
Historia de usuario 4

HISTORIA DE USUARIO	
Número: 4	Usuario: Administrador
Nombre historia: Ventas	
Prioridad: Baja	Riesgo: Alto
Estimación: 4	Iteración: 1
Responsable: Darwin Tipan	
Descripción: La administradora debe poder verificar las acciones realizadas al vender en ventas.	
Detalle: La acción de ventas debe tener las siguientes características:	
<ul style="list-style-type: none">- Formularios donde podrá observar y realizar acciones sobre las ventas realizadas, ventas a crédito, Por entregar, por cobrar, ventas canceladas, cotizaciones.- Generarán reportes en formato PDF por separado.	

Tabla 52.
Historia de usuario 5

HISTORIA DE USUARIO	
Número: 5	Usuario: Administrador
Nombre historia: Compras	
Prioridad: Baja	Riesgo: Alto
Estimación: 5	Iteración: 1
Responsable: Darwin Tipan	
Descripción: La administradora podrá realizar la compra al igual que podrá visualizarla, conjuntamente con las que serán por recibir, pagar y compras canceladas.	
Detalle: El módulo debe tener las siguientes funcionalidades	
<ul style="list-style-type: none">- Se podrá buscar por nombre o código, se podrá añadir nuevas cantidades al inventario- Se podrá generar reportes individuales en formato PDF	

Tabla 53.
Historia de usuario 6

HISTORIA DE USUARIO	
Número: 6	Usuario: Administrador
Nombre historia: Catálogos	
Prioridad: Baja	Riesgo: Alto
Estimación: 6	Iteración: 1
Responsable: Darwin Tipan	
Descripción: La administradora debe poder visualizar y tener acción sobre los Productos, Categorías, Marcas, Clientes, Proveedores, sucursales, administración de precios.	
Detalle: se componen en:	
<ul style="list-style-type: none">- Se podrá visualizar los productos registrados y tener acción sobre estos.- se podrá añadir nuevas categorías, marcas, clientes, proveedores, sucursales se deberá poder administrar precios de estas.- Se podrá generar reportes individuales en formato PDF.	

Tabla 54.
Historia de usuario 7

HISTORIA DE USUARIO	
Número: 7	Usuario: Administrador
Nombre historia: Herramientas	
Prioridad: Baja	Riesgo: Alto
Estimación: 7	Iteración: 1
Responsable: Darwin Tipan	
Descripción: La administradora deberá poder visualizar y realizar acciones sobre la lista de los contactos al igual que podrá comunicarse con usuarios del sistema.	
Detalle: La gestión de los contactos tendrá las siguientes acciones.	
<ul style="list-style-type: none">- Un formulario donde el administrador podrá añadir, editar y eliminar.- Un formulario para comunicarse entre los usuarios registrados	

Tabla 55.
Historia de usuario 8

HISTORIA DE USUARIO	
Número: 8	Usuario: Administrador
Nombre historia: Finanzas	
Prioridad: Baja	Riesgo: Alto
Estimación: 8	Iteración: 1
Responsable: Darwin Tipan	
Descripción: La administradora deberá poder visualizar y realizar acciones sobre el crédito, balance, gastos, caja.	
Detalle: El apartado de finanzas tendrá las siguientes acciones.	
<ul style="list-style-type: none">- Un formulario para el listado de clientes con crédito.- Se deberá generar un balance escogiendo el rango entre fechas específicas.- Un formulario para el listado de gastos- Una lista para las ventas realizadas- Un botón para procesar las ventas, y generar reporte además del historial de cada venta.	

Tabla 56.
Historia de usuario 9

HISTORIA DE USUARIO	
Número: 9	Usuario: Administrador
Nombre historia: Inventario	
Prioridad: Alta	Riesgo: Alto
Estimación: 9	Iteración: 1
Responsable: Darwin Tipan	
Descripción: El administrador podrá controlar el inventario con sus respectivas funciones.	
Detalle: Este módulo contendrá lo siguiente:	

- Un inventario principal, donde deberá mostrar todos los productos registrados,
- Se añadirán botones de acceso rápido que llevarán a la ventana principal y almacenes.
- Listado de inventarios individuales por sucursal, donde podrá editar eliminar y añadir nuevas sucursales.
- Generará historial de cada producto registrado.
- Contendrá un abastecimiento de inventario, permitiendo buscar por código o por nombre al igual que los traspasos y devoluciones.
- Se mostrará el valor del inventario.
- Generará reportes en formato PDF

Tabla 57.
Historia de usuario 10

HISTORIA DE USUARIO	
Número: 10	Usuario: Administrador
Nombre historia: Reportes y administración	
Prioridad: Baja	Riesgo: Alto
Estimación: 10	Iteración: 1
Responsable: Darwin Tipan	
Descripción: El administrador deberá poder generar los reportes de todo el sistema, además de poder configurarlo.	
Detalle: El formulario deberá tener las siguientes funcionalidades al igual que se podrá tener configuraciones básicas.	
<ul style="list-style-type: none"> - Seleccionar la sucursal, producto, fecha “Reporte de inventario” - Seleccionar la sucursal, categoría, fecha “Reporte por categorías” - Seleccionar el vendedor, cliente, fecha “Reporte de ventas” - Seleccionar todas las compras, fecha “Reporte de compras” - Seleccionar todos los pagos, fecha “Reporte de pagos” - Seleccionar el rango de fecha” Reporte de clientes populares” - Seleccionar el rango de fecha “Reporte de vendedores populares” - Seleccionar el rango de fecha “Reporte de productos populares” 	

- Contendrá la lista de los usuarios con sus respectivos roles.
- Se añadirán botones para sus respectivas funciones.
- Se colocarán campos donde podrán tener una configuración básica.
- Se añadirá una función para poder importar datos en archivo. Csv

Tabla 58.

Historia de usuario 11

HISTORIA DE USUARIO

Número:11

Usuario: vendedor

Nombre historia: vendedor

Prioridad: Baja

Riesgo: Bajo

Estimación: 11

Iteración: 1

Responsable: Darwin Tipan

Descripción: En este módulo el vendedor deberá poder observar y manipular las siguientes ventanas.

Detalle: Este módulo deberá tener lo siguiente:

- Deberá poder visualizar un listado donde se pueda buscar por código o nombre para poder realizar la venta.
 - También deberá tener ventas por entregar, cobrar, ventas canceladas y cotizaciones.
 - Podrá observar el inventario principal de la sucursal que se le haya asignado al momento de generar el usuario y contraseña.
 - Tendrá la opción de buscar productos por sucursales para verificar que se encuentre en stock.
 - Podrá realizar peticiones a esa sucursal que se le sea asignado ese producto a la sucursal que la está solicitando.
 - También podrá realizar devoluciones, se tendrá que enviar la devolución al administrador de la venta realizada.
 - Se podrá comunicar con dicho administrador para informar sobre las devoluciones que se realicen.
-

Tabla 59.
Historia de usuario 12

HISTORIA DE USUARIO	
Número: 12	Usuario: Bodeguero
Nombre historia: Bodeguero	
Prioridad: Baja	Riesgo: Bajo
Estimación: 12	Iteración: 1
Responsable: Darwin Tipan	
Descripción: En este módulo el bodeguero deberá poder observar y manipular las siguientes ventanas.	
Detalle: Este módulo deberá tener lo siguiente:	
<ul style="list-style-type: none">- Se deberá mostrar la cantidad de productos registrados, los clientes, ventas e ingresos del mes.- Se le permitirá vender los productos que se encuentren registrados y los nuevos que se estén registrando en ese momento.- Se le permitirá visualizar y manejar las ventas por entregar, por cobrar, ventas canceladas y las cotizaciones que se puedan realizar mientras esté registrando los productos.- Este deberá tener potestad para realizar las compras al proveedor.- Deberá poder visualizar las compras realizadas donde se mostrará un listado pudiendo descargarlas en PDF, al igual que podrá generar un ticket de la compra realizada.- Se deberá mostrar el inventario principal donde se indique la disponibilidad, por entregar, por recibir.- Este podrá abastecer a la sucursal que se le sea asignada al igual que a las demás.	

Tareas de ingeniería.

Tabla 60.

Tarea de usuario 1

Tarea de usuario	
Número de la tarea: 1	Número de historia: 1
Nombre de tarea: Crear el modelo de datos de usuario.	
Tipo de tarea: Desarrollo	
Fecha inicio: 5/02/2021	Fecha fin: 10/02/2021
Programador responsable: Darwin Tipan	
Descripción: Codificar el modelo de inicio de sesión utilizando el framework php codeigniter versión 4 en el entorno de desarrollo Vs Code .	

Tabla 61.

Tarea de usuario 2

Tarea de usuario	
Número de la tarea: 2	Número de historia: 1
Nombre de tarea: Diseñar las interfaces del sistema web.	
Tipo de tarea: Desarrollo	
Fecha inicio: 15/02/2021	Fecha fin: 20/02/2021
Programador responsable: Darwin Tipan	
Descripción: Codificar las interfaces dentro template con Bootstrap y librerías de HTML5, Css, JavaScript.	

Tabla 62.*Tarea de usuario 3*

Tarea de usuario	
Número de la tarea: 3	Número de historia: 1
Nombre de tarea: Validar datos de inicio de sesión.	
Tipo de tarea: Desarrollo	
Fecha inicio: 21/02/2021	Fecha fin: 21/02/2021
Programador responsable: Darwin Tipan	
Descripción: Programar la sentencia lógica dentro del entorno vs code y validar la existencia en la base de datos.	

Tabla 63.*Tarea de usuario 4*

Tarea de usuario	
Número de la tarea: 4	Número de historia: 1
Nombre de tarea: Codificar vistas de acceso de usuarios.	
Tipo de tarea: Desarrollo	
Fecha inicio: 22/02/2021	Fecha fin: 26/02/2021
Programador responsable: Darwin Tipan	
Descripción: Programar la sentencia lógica de envío de datos así el template	

Tabla 64.*Tarea de usuario 5*

Tarea de usuario	
Número de la tarea: 5	Número de historia: 1
Nombre de tarea: Crear modelos de datos. Roles, permisos, logs de usuario.	
Tipo de tarea: Desarrollo	
Fecha inicio: 1/03/2021	Fecha fin: 5/03/2021
Programador responsable: Darwin Tipan	
Descripción: Programar la sentencia lógica para los modelos y envío de datos dentro del entorno de programación y validar su funcionamiento.	

Tabla 65.*Tarea de usuario 6*

Tarea de usuario	
Número de la tarea: 6	Número de historia: 1
Nombre de tarea: Diseñar las vistas del administrador.	
Tipo de tarea: Desarrollo	
Fecha inicio: 6/03/2021	Fecha fin: 10/03/2021
Programador responsable: Darwin Tipan	
Descripción: Programar las sentencias de las vistas utilizando Bootstrap con librerías de JavaScript y Css.	

Tabla 66.

Tarea de usuario 7

Tarea de usuario	
Número de la tarea: 7	Número de historia: 1
Nombre de tarea: Codificar las vistas del administrador.	
Tipo de tarea: Desarrollo	
Fecha inicio: 11/03/2021	Fecha fin: 15/03/2021
Programador responsable: Darwin Tipan	
Descripción: Programar las sentencias de las vistas utilizando Bootstrap con librerías de JavaScript y Css.	

Tabla 67.

Tarea de usuario 8

Tarea de usuario	
Número de la tarea: 8	Número de historia: 1
Nombre de tarea: Diseñar las interfaces de roles y las diferentes vistas de los usuarios creados dentro del template.	
Tipo de tarea: Desarrollo	
Fecha inicio: 16/03/2021	Fecha fin: 20/03/2021
Programador responsable: Darwin Tipan	
Descripción: Programar las vistas y las interfaces que contengan las sentencias de envío de datos desde el template al gestor phpMyAdmin en el entorno de desarrollo vs code.	

Tabla 68.

Tarea de usuario 9

Tarea de usuario	
Número de la tarea: 9	Número de historia: 1
Nombre de tarea: Codificar el envío de datos a los formularios	
Tipo de tarea: Desarrollo	
Fecha inicio: 21/03/2021	Fecha fin: 28/03/2021
Programador responsable: Darwin Tipan	
Descripción: Programar las clases y sentencias lógicas para el envío de datos	

Tabla 69.

Tarea de usuario 10

Tarea de usuario	
Número de la tarea: 10	Número de historia: 1
Nombre de tarea: Migración de vistas y sentencias al template	
Tipo de tarea: Desarrollo	
Fecha inicio: 2/04/2021	Fecha fin: 20/04/2021
Programador responsable: Darwin Tipan	
Descripción: Codificar las rutas de envío de las vistas y sentencias al template	

Tabla 70.
Tarea de usuario 11

Tarea de usuario	
Número de la tarea: 11	Número de historia: 1
Nombre de tarea: Codificar las vistas de gestión de control de usuarios.	
Tipo de tarea: Desarrollo	
Fecha inicio: 21/04/2021	Fecha fin: 28/04/2021
Programador responsable: Darwin Tipan	
Descripción: Programar las vistas que contenga las clases y métodos de roles, permisos y logs de usuario que permitan renderizar los datos hacia el template	

Tabla 71.
Tarea de usuario 12

Tarea de usuario	
Número de la tarea: 12	Número de historia: 1
Nombre de tarea: Codificar las vistas de los usuarios creados.	
Tipo de tarea: Desarrollo	
Fecha inicio: 1/05/2021	Fecha fin: 15/05/2021
Programador responsable: Darwin Tipan	
Descripción: Crear las ventanas que pertenecen a cada usuario creado utilizando HTML5 y Bootstrap.	

Tabla 72.
Tarea de usuario 13

Tarea de usuario	
Número de la tarea: 13	Número de historia: 1
Nombre de tarea: Crear modelo de datos de ventas, compras, catálogos, herramientas, finanzas, inventario, reportes, administración	
Tipo de tarea: Desarrollo	
Fecha inicio: 16/05/2021	Fecha fin: 22/05/2021
Programador responsable: Darwin Tipan	
Descripción: programar el modelo de datos y clases de ventas, compras, catálogos, administración conjuntamente con sus permisos y logs de usuario que permitan renderizar los datos hacia el template	

Tabla 73.
Tarea de usuario 14

Tarea de usuario	
Número de la tarea: 14	Número de historia: 1
Nombre de tarea: Crear el modelo de datos de ventas de los sub menús. Ventas, ventas crédito, por entregar, por cobrar, ventas canceladas, cotizaciones.	
Tipo de tarea: Desarrollo	
Fecha inicio: 23/05/2021	Fecha fin: 28/05/2021
Programador responsable: Darwin Tipan	
Descripción: Programar el modelo de vista que contenga las clases y métodos del módulo de ventas que permitan renderizar los datos hacia el template	

Tabla 74.
Tarea de usuario 15

Tarea de usuario	
Número de la tarea: 15	Número de historia: 1
Nombre de tarea: Crear el modelo de datos de compras de los sub menús. Nueva*, compras, por recibir, por pagar, compras canceladas.	
Tipo de tarea: Desarrollo	
Fecha inicio: 01/06/2021	Fecha fin: 03/06/2021
Programador responsable: Darwin Tipan	
Descripción: Programar el modelo de vista que contenga las clases y métodos del módulo de compras que permitan renderizar los datos hacia el template	

Tabla 75.
Tarea de usuario 16

Tarea de usuario	
Número de la tarea: 16	Número de historia: 1
Nombre de tarea: Crear el modelo de datos de catálogos de los sub menús. Productos, categorías, marcas, cliente, proveedor, sucursales, administrar precios.	
Tipo de tarea: Desarrollo	
Fecha inicio: 06/06/2021	Fecha fin: 08/06/2021
Programador responsable: Darwin Tipan	
Descripción: Programar el modelo de vista que contenga las clases y métodos del módulo de categorías que permitan renderizar los datos hacia el template	

Tabla 76.

Tarea de usuario 17

Tarea de usuario	
Número de la tarea: 17	Número de historia: 1
Nombre de tarea: Crear el modelo de datos de herramientas de los sub menús. Contactos, mensajes.	
Tipo de tarea: Desarrollo	
Fecha inicio: 08/06/2021	Fecha fin: 09/06/2021
Programador responsable: Darwin Tipan	
Descripción: Programar el modelo de vista que contenga las clases y métodos del módulo de herramientas que permitan renderizar los datos hacia el template	

Tabla 77.

Tarea de usuario 18

Tarea de usuario	
Número de la tarea: 18	Número de historia: 1
Nombre de tarea: Crear el modelo de datos de finanzas de los sub menús. Crédito, balance, gastos, caja chica, caja.	
Tipo de tarea: Desarrollo	
Fecha inicio: 10/06/2021	Fecha fin: 12/06/2021
Programador responsable: Darwin Tipan	
Descripción: Programar el modelo de vista que contenga las clases y métodos del módulo de finanzas que permitan renderizar los datos hacia el template	

Tabla 78.
Tarea de usuario 19

Tarea de usuario	
Número de la tarea: 19	Número de historia: 1
Nombre de tarea: Crear el modelo de datos de inventario de los sub menús. Inventario principal, abastecer, valor del inventario, buscar productos, inventario global, traspasar, devolución, traspasos, devolución.	
Tipo de tarea: Desarrollo	
Fecha inicio: 13/06/2021	Fecha fin: 22/06/2021
Programador responsable: Darwin Tipan	
Descripción: Programar el modelo de vista que contenga las clases y métodos del módulo de control de inventario que permitan renderizar los datos hacia el template.	

Tabla 79.
Tarea de usuario 20

Tarea de usuario	
Número de la tarea: 20	Número de historia: 1
Nombre de tarea: Crear el modelo de datos de reportes de los sub menús.log de inventario, por categorías, ventas, compras, reporte de pagos, clientes populares, vendedores populares, productos populares.	
Tipo de tarea: Desarrollo	
Fecha inicio: 22/06/2021	Fecha fin: 26/06/2021
Programador responsable: Darwin Tipan	
Descripción: Programar el modelo de vista que contenga las clases y métodos del módulo de reportes que permitan renderizar los datos hacia el template	

Tabla 80.
Tarea de usuario 21

Tarea de usuario	
Número de la tarea: 21	Número de historia: 1
Nombre de tarea: Crear el modelo de datos de administración de los sub menús, usuarios, configuración, importar datos.	
Tipo de tarea: Desarrollo	
Fecha inicio: 26/06/2021	Fecha fin: 28/06/2021
Programador responsable: Darwin Tipan	
Descripción: Programar el modelo de vista que contenga las clases y métodos del módulo de administración que permitan renderizar los datos hacia el template	

Tabla 81.
Tarea de usuario 22

Tarea de usuario	
Número de la tarea: 22	Número de historia: 1
Nombre de tarea: Creación de método para asignar cuentas a los usuarios creados.	
Tipo de tarea: Desarrollo	
Fecha inicio: 01/07/2021	Fecha fin: 05/07/2021
Programador responsable: Darwin Tipan	
Descripción: Se debe crear un método específico, que permitirá la asignación de cada ventana y pestaña a los diferentes roles.	

Tabla 82.*Tarea de usuario 23*

Tarea de usuario	
Número de la tarea: 23	Número de historia: 1
Nombre de tarea: Creación de vistas, vendedor, bodeguero, administrador.	
Tipo de tarea: Desarrollo	
Fecha inicio: 01/07/2021	Fecha fin: 05/07/2021
Programador responsable: Darwin Tipan	
Descripción: Se debe crear un método específico para las vistas que permitirá la asignación de cada ventana.	

Tabla 83.*Tarea de usuario 24*

Tarea de usuario	
Número de la tarea: 24	Número de historia: 1
Nombre de tarea: Generar documentos y reportes.	
Tipo de tarea: Desarrollo	
Fecha inicio: 06/07/2021	Fecha fin: 07/07/2021
Programador responsable: Darwin Tipan	
Descripción: Programar en el template la funcionalidad de imprimir un documento PDF generado por datos ingresados en el sistema.	

Tabla 84.*Tarea de usuario 25*

Tarea de usuario

Número de la tarea: 25**Número de historia:** 1**Nombre de tarea:** Generador de barra estadística en el template.**Tipo de tarea:** Desarrollo**Fecha inicio:** 08/07/2021**Fecha fin:** 09/07/2021**Programador responsable:** Darwin Tipan**Descripción:** Programar una barra estadística donde muestre los movimientos de los últimos 30 días en el sistema, esto se logra por la funcionalidad que trae el template.

Tabla 85.*Tarea de usuario 26*

Tarea de usuario

Número de la tarea: 26**Número de historia:** 1**Nombre de tarea:** Generador de complementos Html5, Css, JavaScript.**Tipo de tarea:** Desarrollo**Fecha inicio:** 10/07/2021**Fecha fin:** 11/07/2021**Programador responsable:** Darwin Tipan**Descripción:** descarga de los complementos para las interfaces del template.

Tabla 86.
Tarea de usuario 27

Tarea de usuario		
<hr/>		
Número de la tarea: 27	Número de historia: 1	
Nombre de tarea: Envío de datos desde el template al gestor phpMyAdmin.		
Tipo de tarea: Desarrollo		
Fecha inicio: 11/07/2021	Fecha fin: 12/07/2021	
Programador responsable: Darwin Tipan		
Descripción: Sobrescribir los datos en Codeigniter V4 y realizar su respectiva migración		
	hacia	MySQL

Fase de diseño

La fase de diseño se compone de varios elementos. Para empezar, están las tarjetas CRC, que son herramientas de la metodología XP que servían para desglosar las clases o entidades del sistema y entender cuáles son sus responsabilidades y cómo se relacionan entre sí. También está el diagrama de relaciones de la base de datos y los diseños de creación de prototipos creados en Balsamiq y al final de la etapa se definió los diagramas de casos de uso graficados en Draw.IO.

Tarjetas CRC.

Tabla 87.
Tarjeta CRC Box/Caja

Tarjeta CRC	
<hr/>	
Nombre: Box /Caja	
Responsabilidades:	Colaboración:
Guardar información de la entrada y salida de los materiales y accesorios.	
Listar la información de caja.	

Tabla 88.

Tarjeta CRC Category/Categoría

Tarjeta CRC	
Nombre: Category/Categoría	
Responsabilidades:	Colaboración:
Guardar la información proporcionada por el usuario	
Mostrar los listados de cada categoría que sea añadida.	
Listar la información de las categorías.	

Tabla 89.

Tarjeta CRC Configuration/ Configuración

Tarjeta CRC	
Nombre: Configuration/ Configuración	
Responsabilidades:	Colaboración:
Guardar la información de configuración básica	
mostrar los datos proporcionados por el usuario.	
Listar la información de configuración.	

Tabla 90.

Tarjeta CRC D estado de entregas

Tarjeta CRC	
Nombre: D estado de entregas	
Responsabilidades:	Colaboración:
Guardar la información de las entregas por	
recibir, por cobrar etc.	
Mostrar los datos almacenados en un listado con sus respectivos	
campos	

Tabla 91.

Tarjeta CRC F Formas de pago

Tarjeta CRC	
Nombre: F Formas de pago	
Responsabilidades:	Colaboración:
Guardar las tres formas de pago que existen en el sistema.	
Guardar los datos almacenados en la forma de pago.	

Tabla 92.

Tarjeta CRC P Estado de los pagos

Tarjeta CRC	
Nombre: P Estado de los pagos	
Responsabilidades:	Colaboración:
Guardar los estados de pago que existen en el sistema como pago pendiente. cancelado, crédito.	
Guardar los datos almacenados en la forma de pago.	

Tabla 93.

Tarjeta CRC Message/ Mensaje

Tarjeta CRC	
Nombre: Message/ Mensaje	
Responsabilidades:	Colaboración:
Guardar todos los mensajes entre sucursales.	
Mostrar todos los mensajes	

Tabla 94.
Tarjeta CRC Operation / Operación

Tarjeta CRC	
Nombre: Operation / Operación	
Responsabilidades:	Colaboración:
Guardar los datos de las entradas, salidas, traspasos, podrá devolver el valor si es una entrega o devolución.	

Tabla 95.
Tarjeta CRC Operation Type/ Tipo de operación

Tarjeta CRC	
Nombre: Operation Type/ Tipo de operación	
Responsabilidades:	Colaboración:
Guardar el tipo de operación que se esté realizando en la tabla de operación.	

Tabla 96.
Tarjeta CRC Payment/ Pago

Tarjeta CRC	
Nombre: Payment/ Pago	
Responsabilidades:	Colaboración:
Guardar los datos relacionados con pagos, ventas, personas, valor, donde también podrá vender a crédito.	

Tabla 97.
Tarjeta CRC Spend/ Gasto

Tarjeta CRC	
Nombre: Spend/ Gasto	Colaboración:
Guardar la información de la caja del administrador, vendedor, bodeguero.	

Tabla 98.*Tarjeta CRC XX/ inicializar el gráfico*

Tarjeta CRC	
Nombre: XX/ inicializar el gráfico positivo	Colaboración:
Guardar los valores para inicializar el gráfico de las ventas.	

Tabla 99.*Tarjeta CRC YY/ inicializar el gráfico*

Tarjeta CRC	
Nombre: YY/ inicializar el gráfico negativo	Colaboración:
Guardar los valores para inicializar el gráfico en las pérdidas.	

Tabla 100.*Tarjeta CRC Saving/ Ahorro*

Tarjeta CRC	
Nombre: Spend/ Gasto	Colaboración:
Guardar los datos de los clientes que opten por tener un crédito.	

Tabla 101.*Tarjeta CRC Sell/ Facturas*

Tarjeta CRC	
Nombre: Sell/ Facturas	Colaboración:
Generar todos los datos para generar el ticket de factura.	

Tabla 102.*Tarjeta CRC Brand/ Marca*

Tarjeta CRC	
Nombre: Brand/ Marca	Colaboración:
Guardar la información de las marcas y categorías.	
Mostrar un listado de todas las marcas y categorías registradas.	

Tabla 103.*Tarjeta CRC Person/Personas*

Tarjeta CRC	
Nombre: Person/Personas	Colaboración:
Guardar todos los datos de los clientes y proveedores.	
Mostrar un listado de los clientes y proveedores registrados	

Tabla 104.*Tarjeta CRC Payment Type/ tipo de pago*

Tarjeta CRC	
Nombre: Payment Type/ tipo de pago	Colaboración:
Guardar los valores de los créditos que se generen	
Mostrar un listado de los clientes con créditos	

Tabla 105.*Tarjeta CRC User/Usuarios*

Tarjeta CRC	
Nombre: User/Usuarios	Colaboración:
Guardar todos los datos de los usuarios creados con sus respectivas delegaciones.	
Mostrar un listado de todos los usuarios donde se podrá editar y eliminar los campos.	

Tabla 106.*Tarjeta CRC Stock/ Existencias*

Tarjeta CRC	
Nombre: Stock/ Existencias	Colaboración:
Guardar los datos relacionados con la tabla de precio, operación, producto.	
Mostrar el listado de datos registrados según el producto registrado.	

Tabla 107. *Tarjeta CRC Price/precio*

Tarjeta CRC	
Nombre: Price/precio	Colaboración:
Guardar los datos relacionados con la tabla de precio, operación, producto.	
Mostrar el listado de datos registrados según el producto.	

Tabla 108.
Tarjeta CRC Product/producto

Tarjeta CRC	
Nombre: Product/producto	Colaboración:
Guardar todos los datos relacionados con el registro de un nuevo M/A	
Listar todos los M/Á registrados.	

Diagrama UML

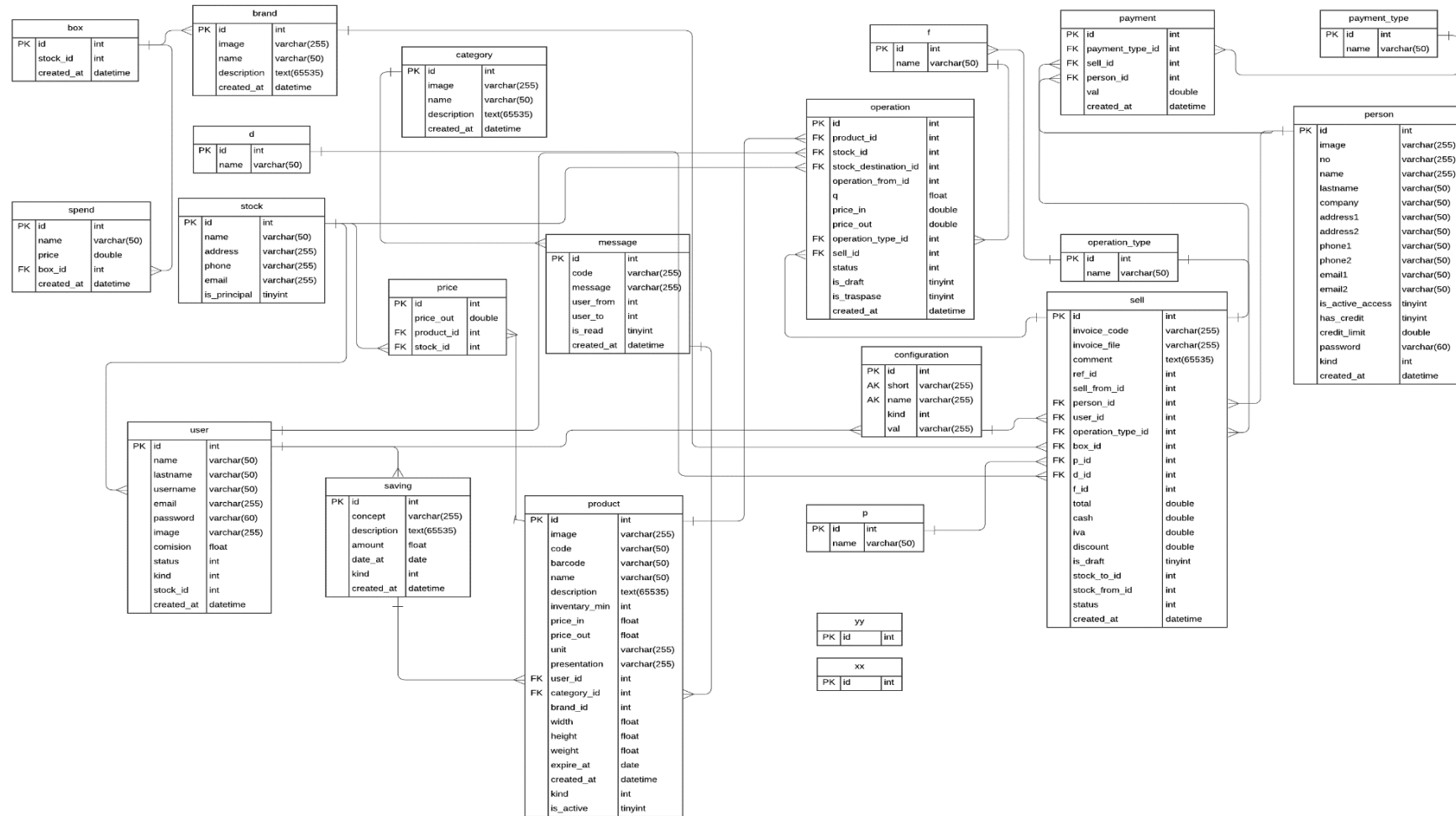


Figura 13. Diseño UML

Diseño de base de datos

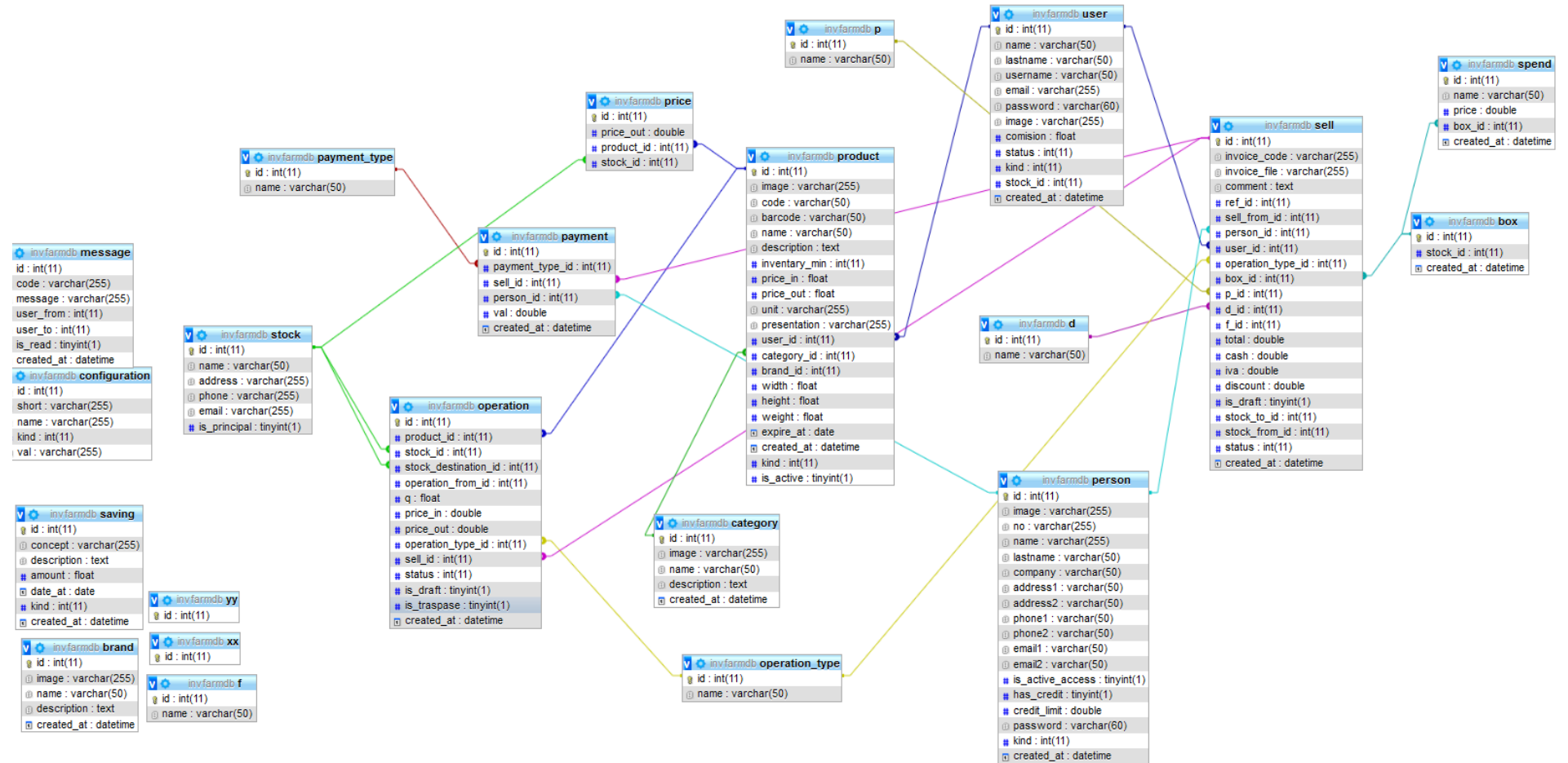


Figura 14. Diseño de base de datos

Diseño de prototipo

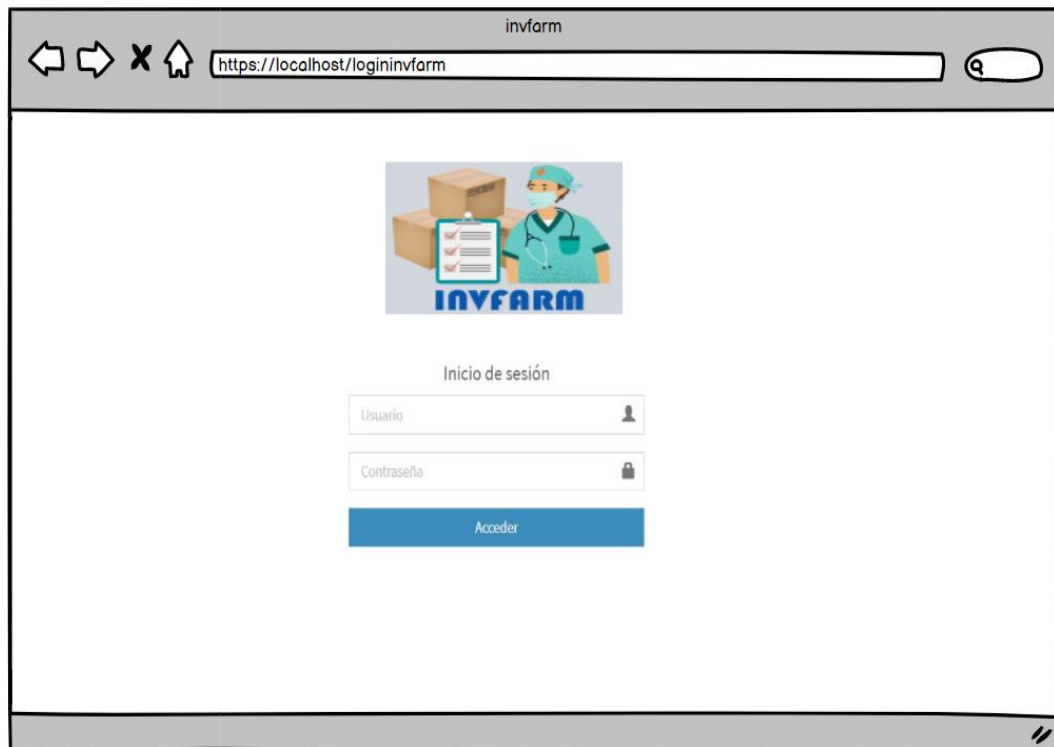


Figura 15. Maquetación de inicio de sesión, administrador

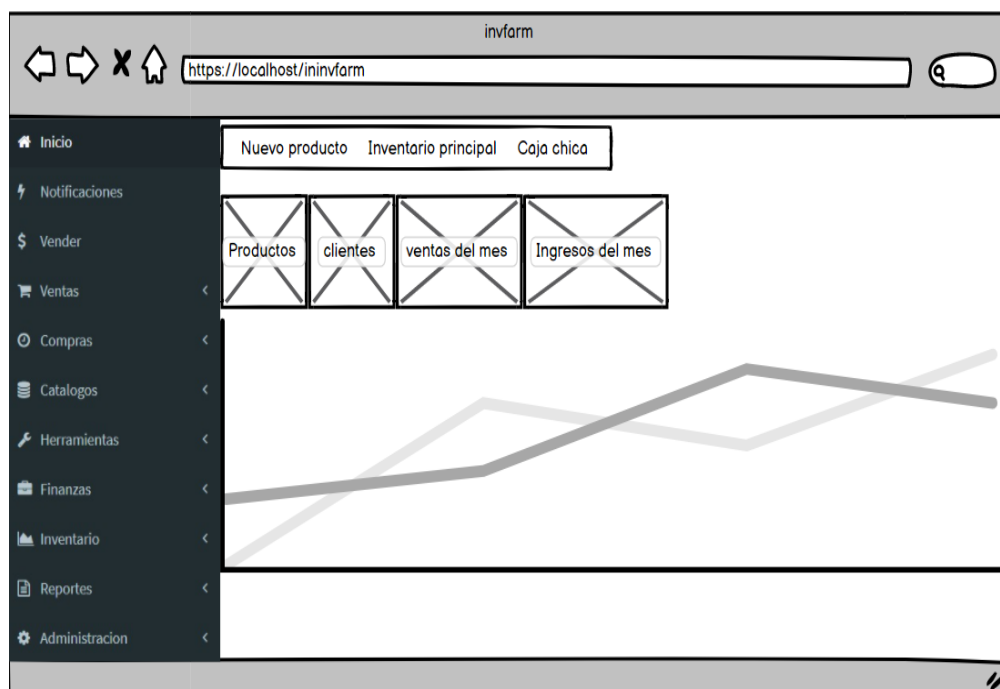


Figura 16. Maquetación de panel de control

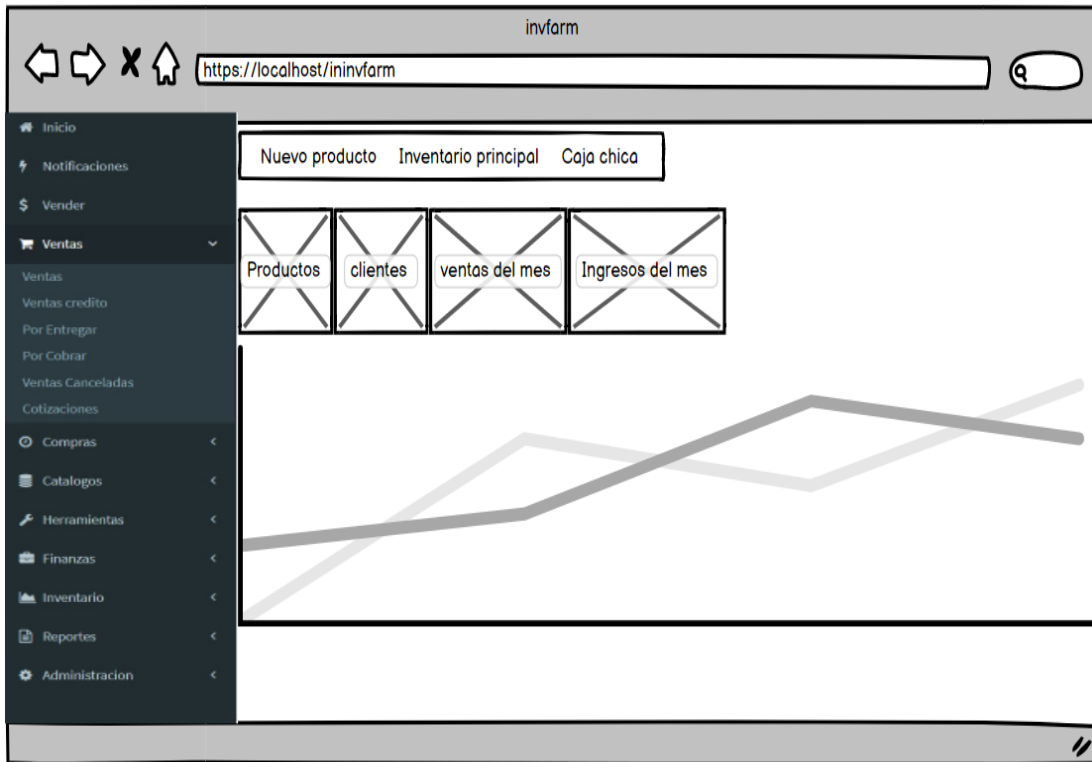


Figura 17. Maquetación de ventas

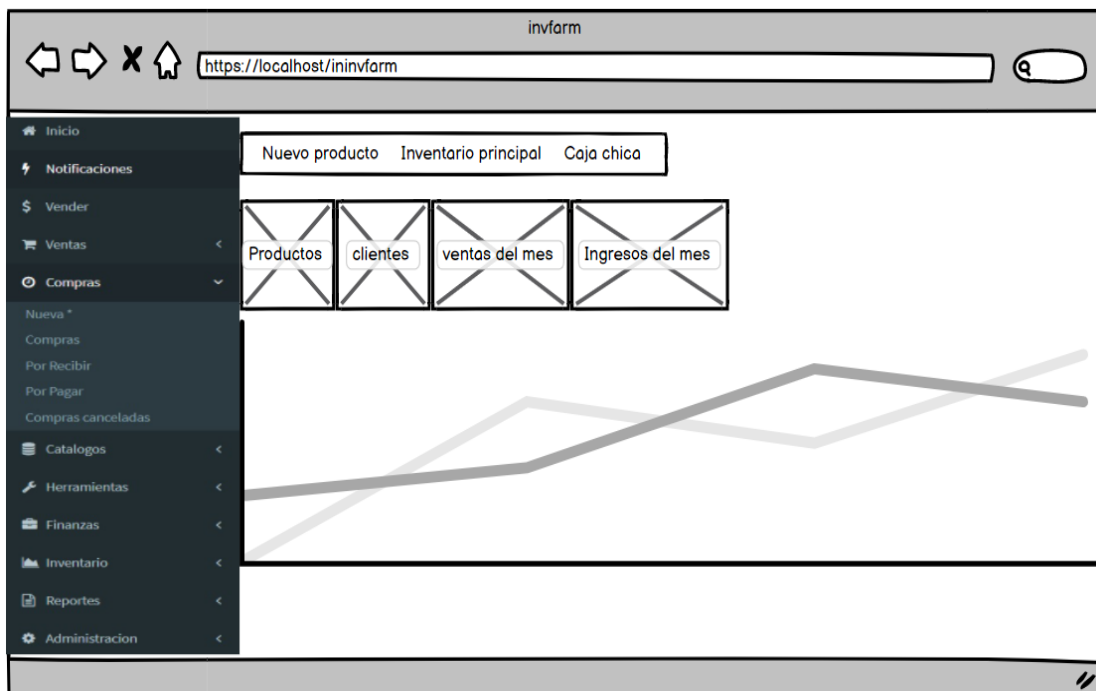


Figura 18. Maquetación de compras

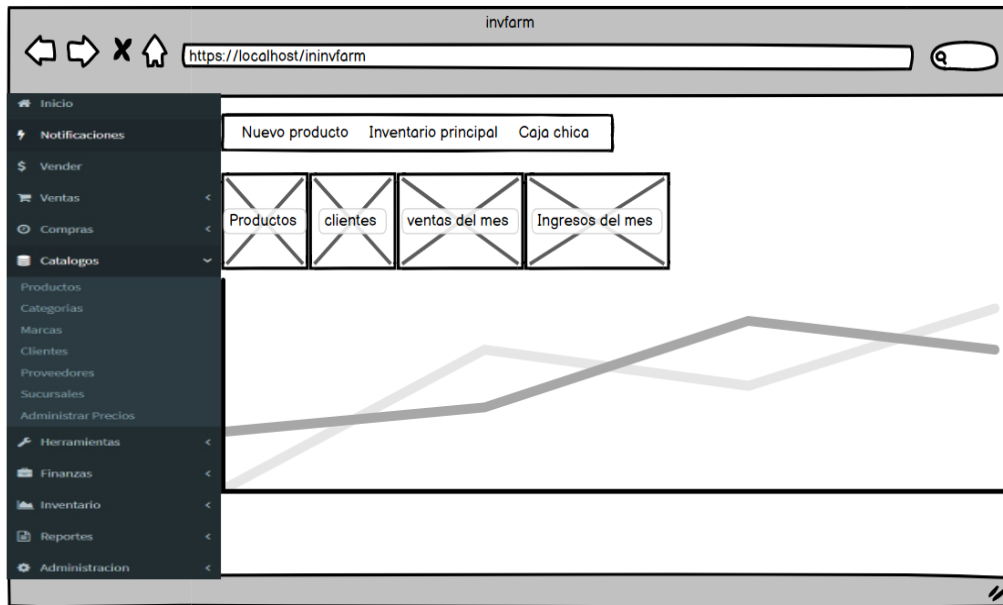


Figura 19. Maquetación de catálogos

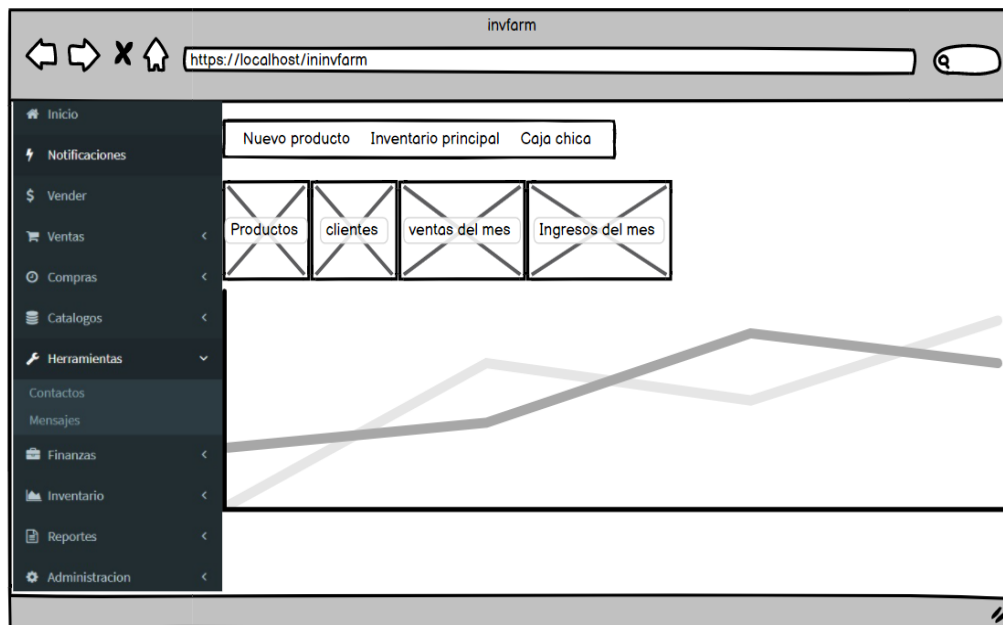


Figura 20. Maquetación de herramientas

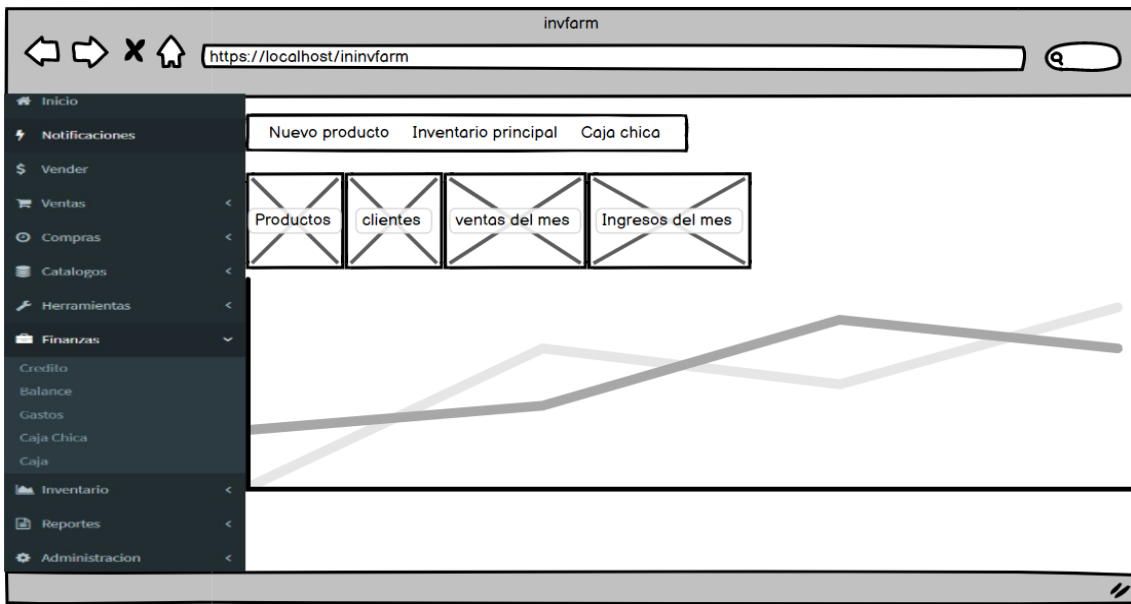


Figura 21. Maquetación de finanzas

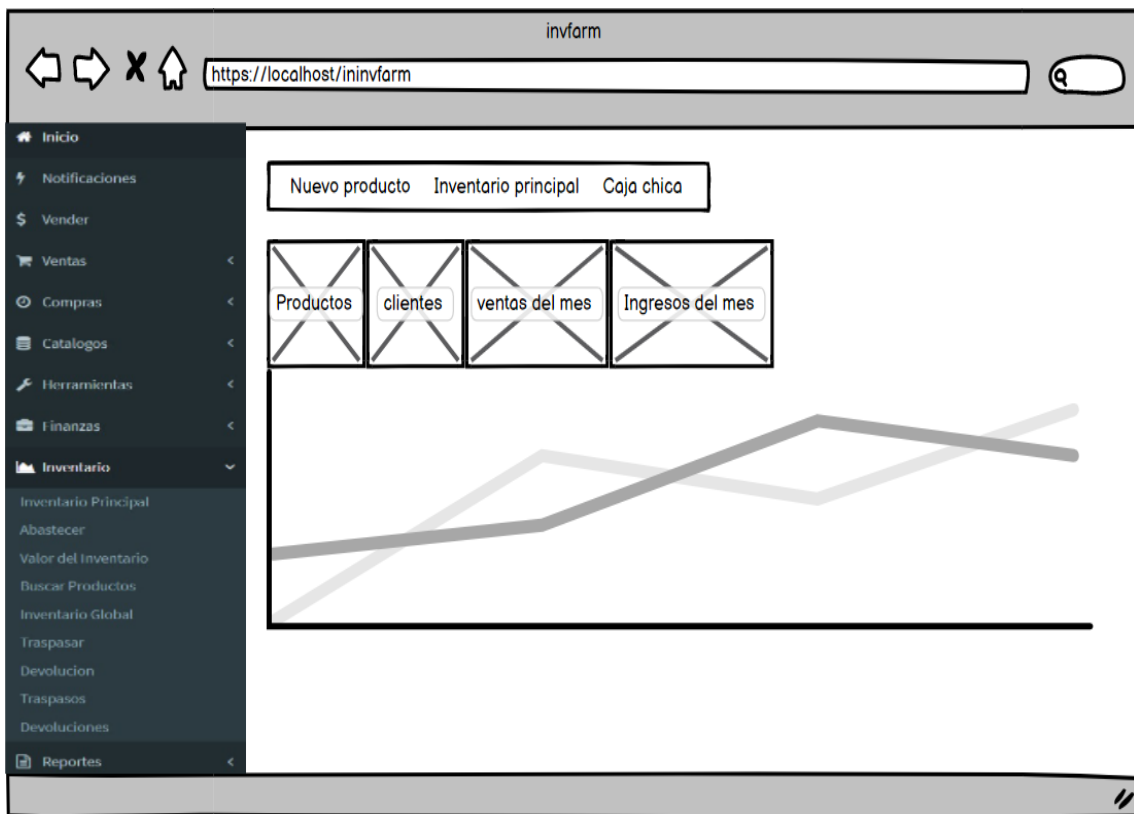


Figura 22. Maquetación de inventario

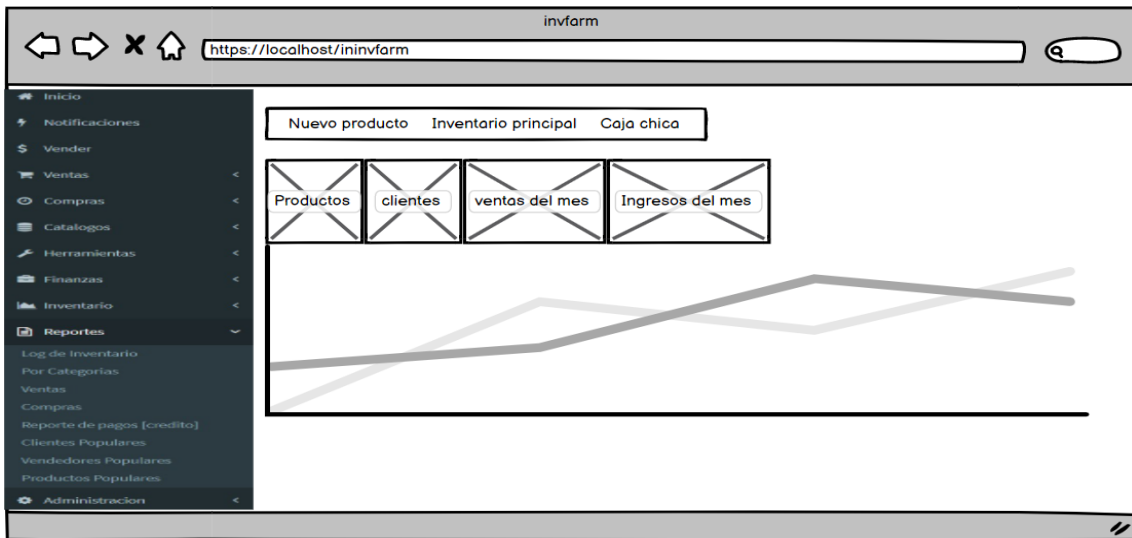


Figura 23. Maquetación de reportes

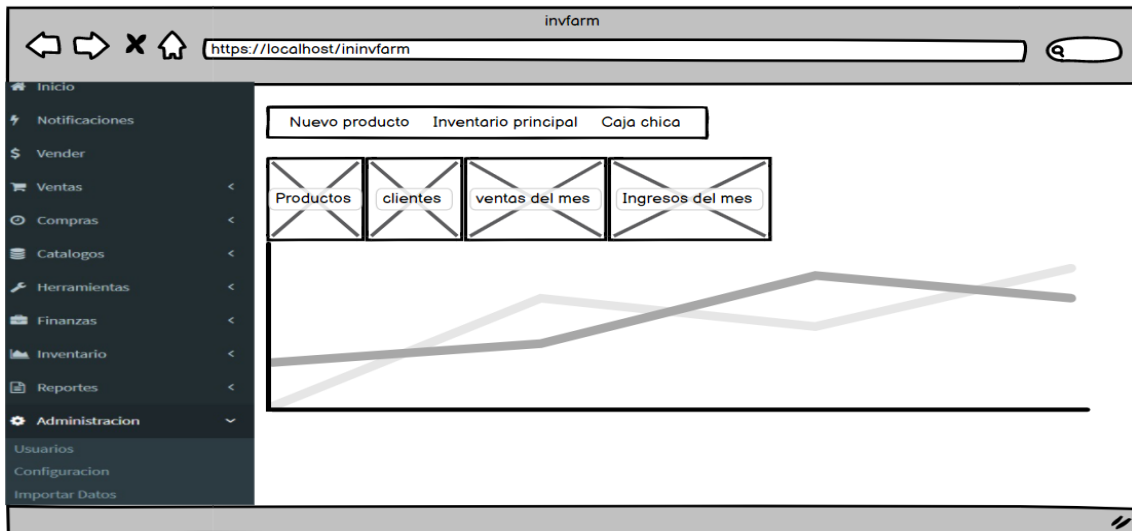


Figura 24. Maquetación de administración

Diseño de prototipos: Bodeguero

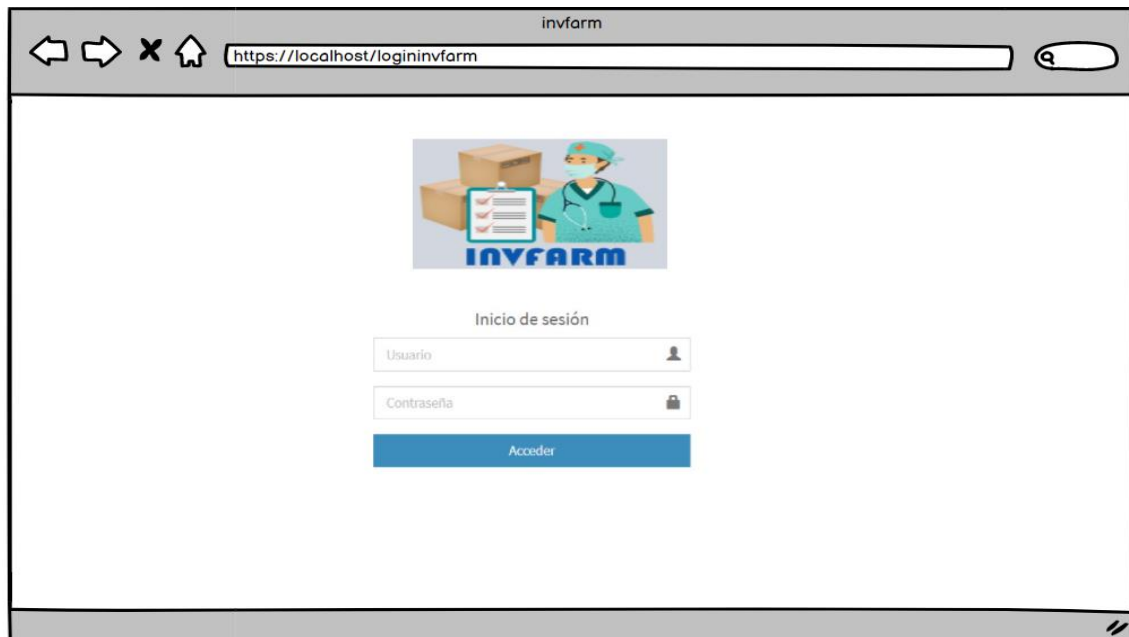


Figura 25. Maquetación de inicio de sesión

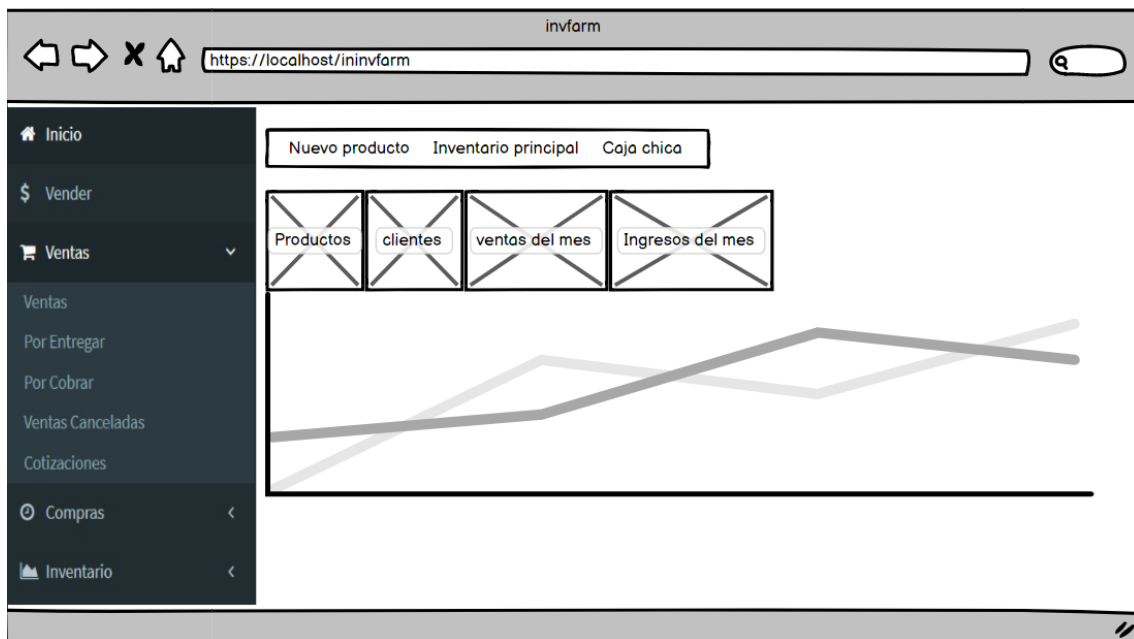


Figura 26. Maquetación de ventas

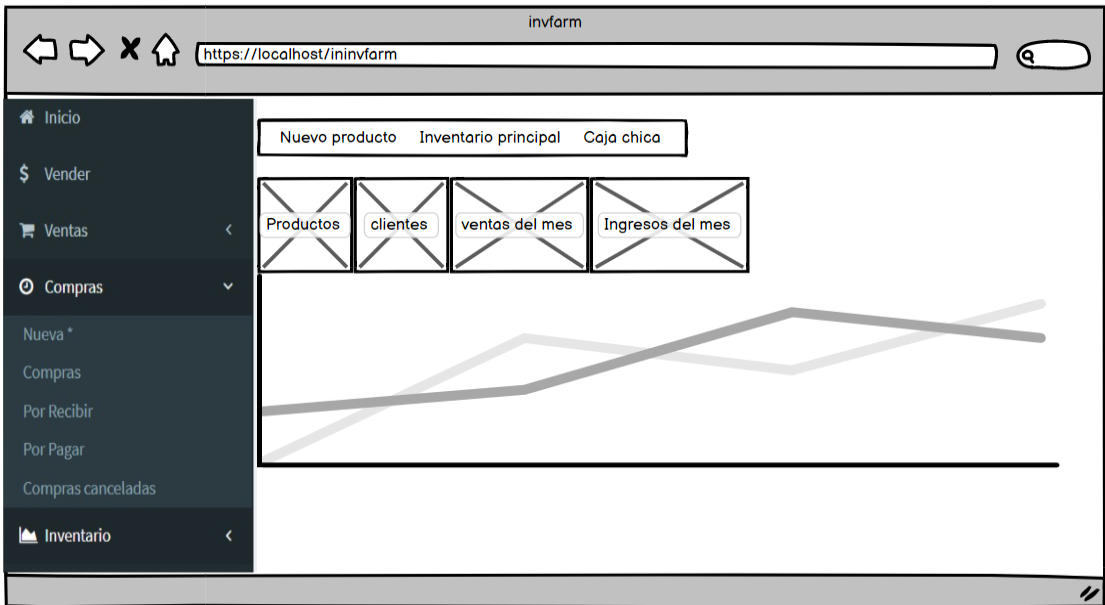


Figura 27. Maquetación de compras

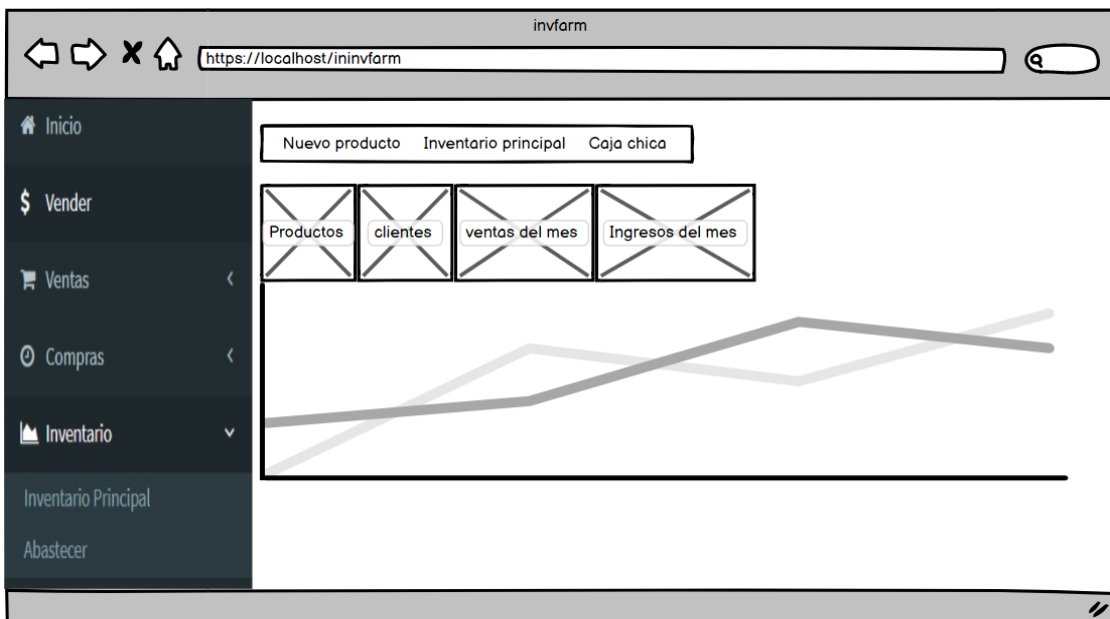


Figura 28. Maquetación de inventario

Diseño de prototipos: Vendedor



Figura 29. Maquetación de inicio de sesión

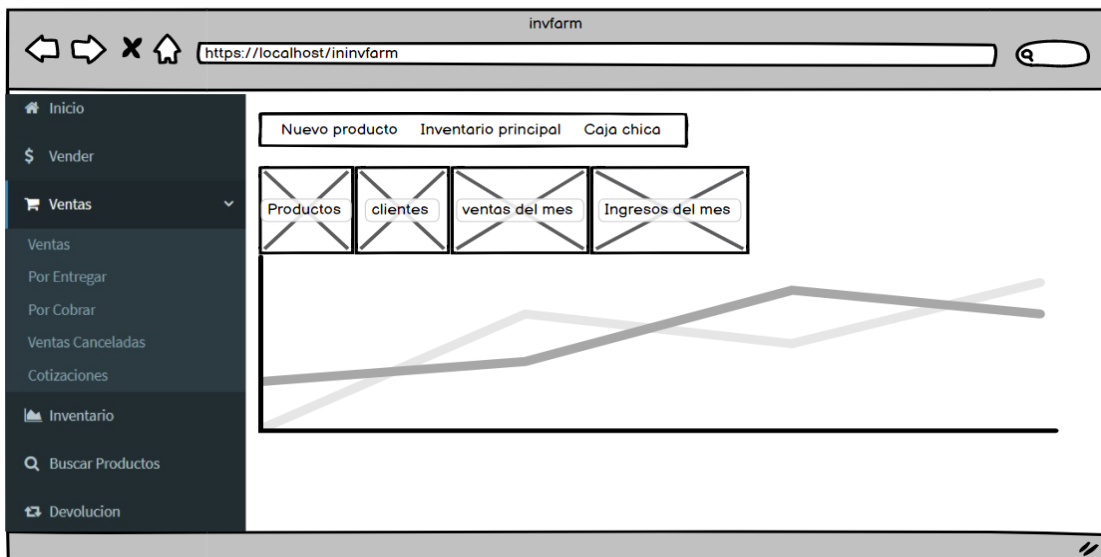


Figura 30. Maquetación de ventas

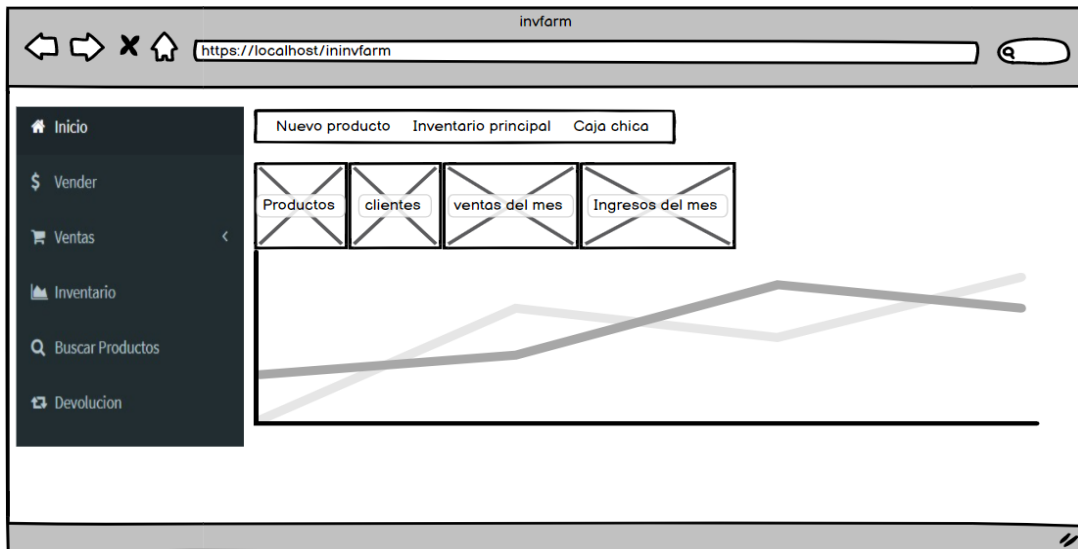


Figura 31. Maquetación de inventario, buscar producto, devolución

Diagramas de casos de uso.

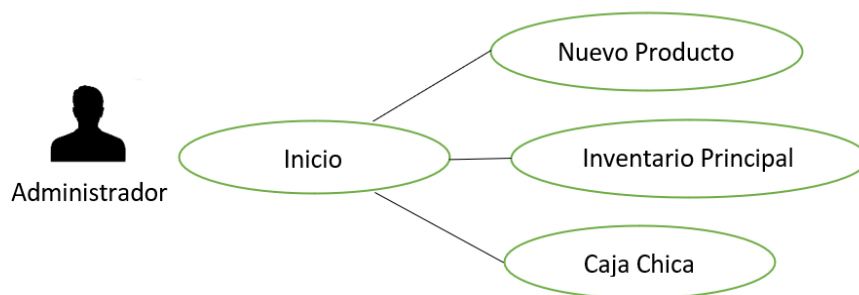


Figura 32. Diagrama de caso de uso administrador/inicio

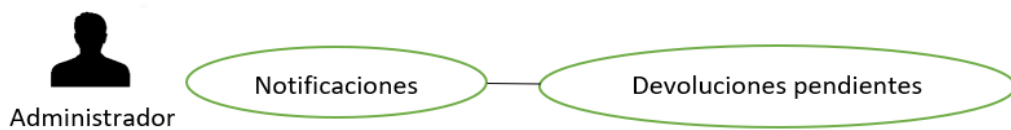


Figura 33. Diagrama de caso de uso administrador/notificaciones

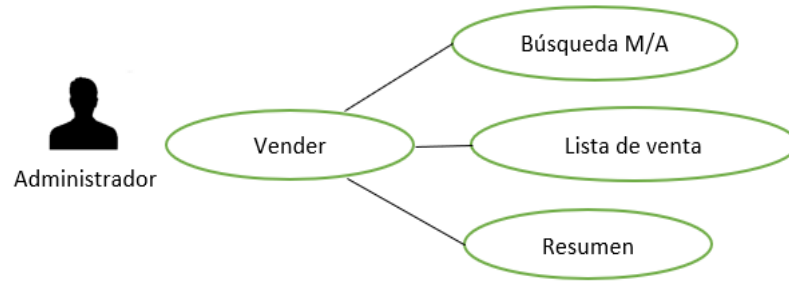


Figura 34. Diagrama de caso de uso administrador/vender

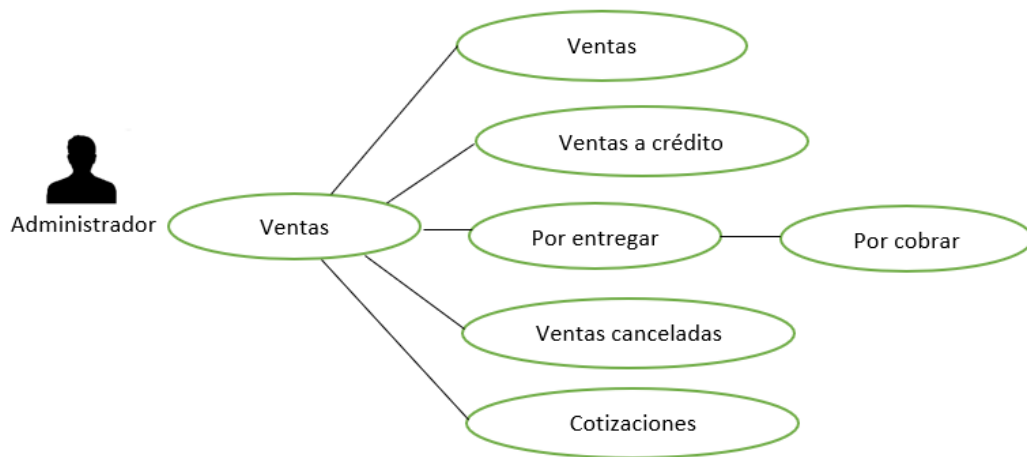


Figura 35. Diagrama de caso de uso administrador/ventas

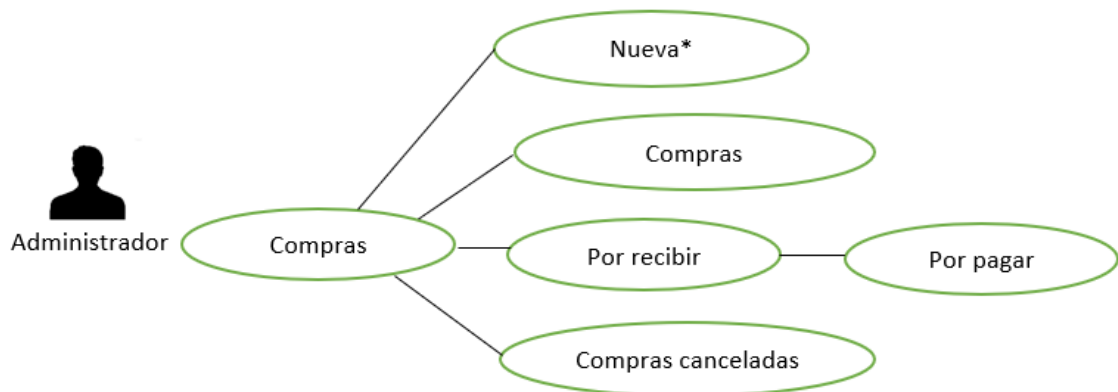


Figura 36. Diagrama de caso de uso administrador/compras



Figura 37. Diagrama de caso de uso administrador/catálogos

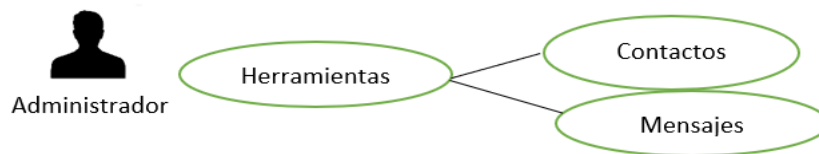


Figura 38. Diagrama de caso de uso administrador/herramientas

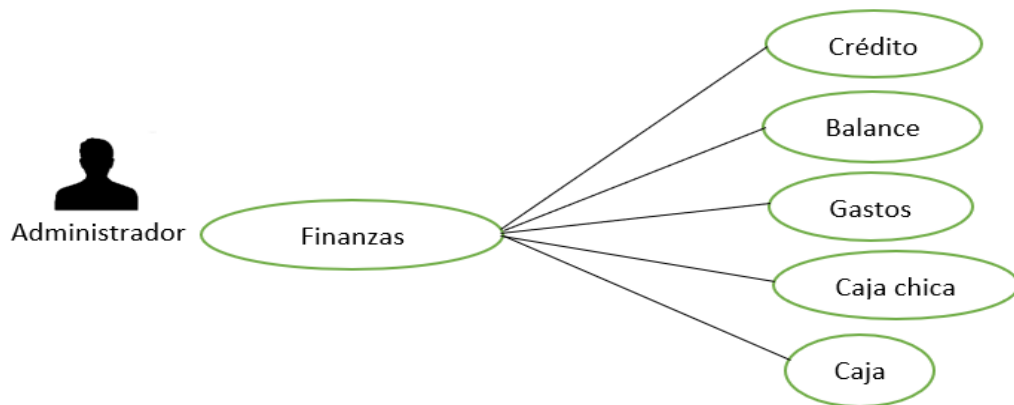


Figura 39. Diagrama de caso de uso administrador/finanzas

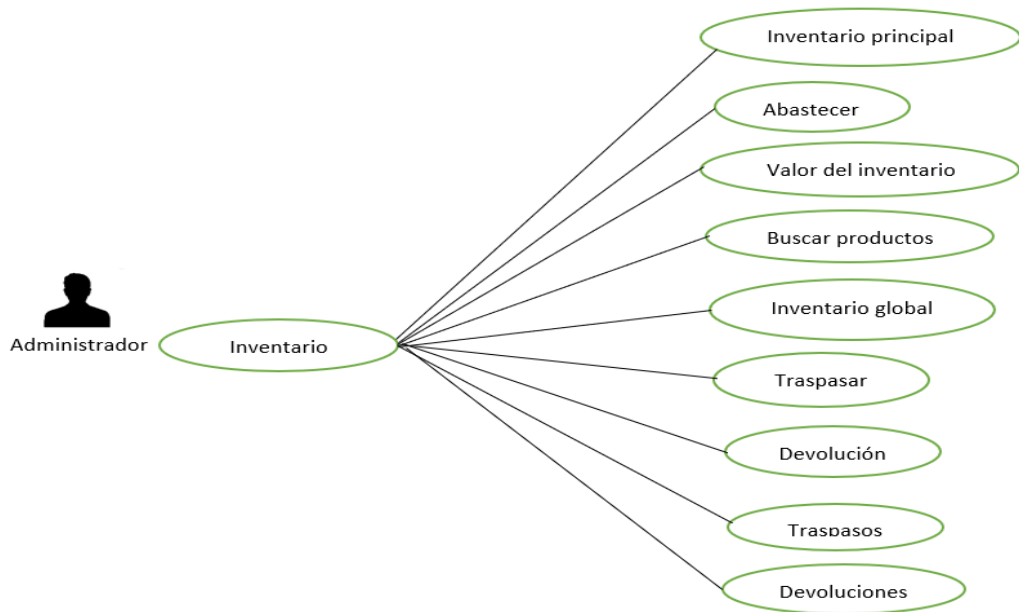


Figura 40. Diagrama de caso de uso administrador/inventario

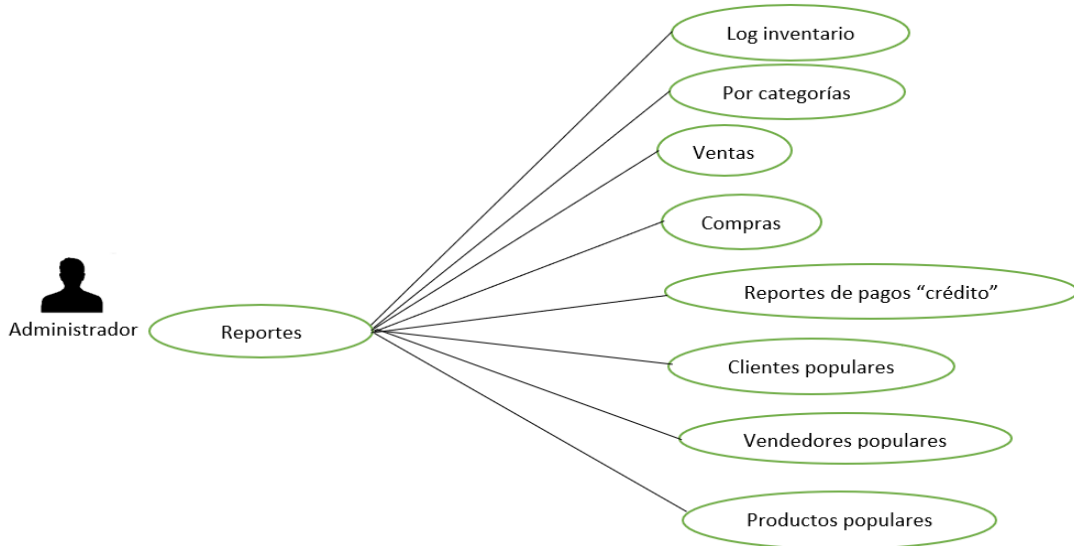


Figura 41. Diagrama de caso de uso administrador/reportes

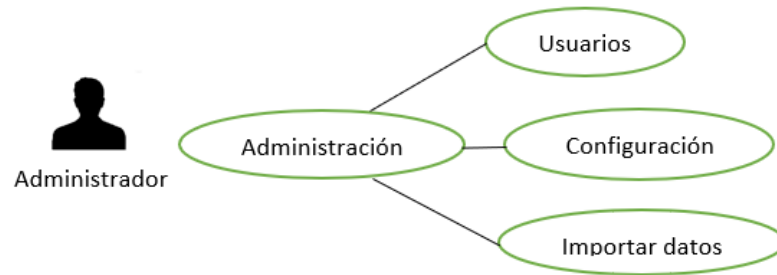


Figura 42. Diagrama de caso de uso administrador/administrador

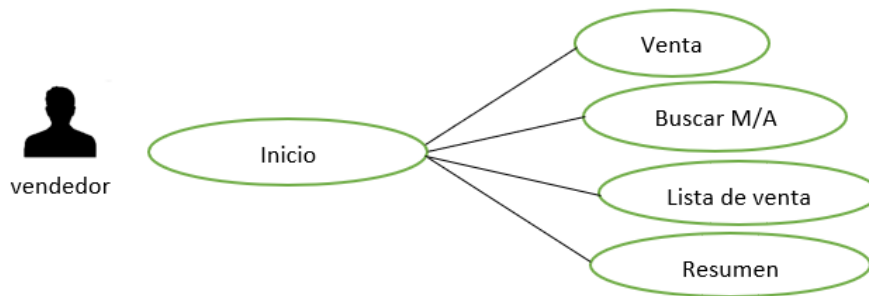


Figura 43. Diagrama de caso de uso vendedor/inicio

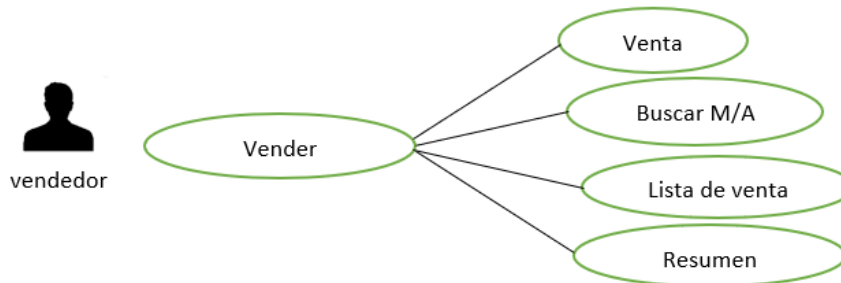


Figura 44. Diagrama de caso de uso vendedor/vender

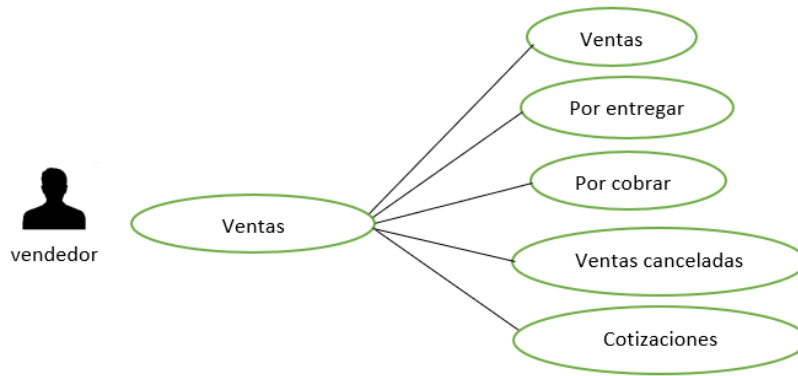


Figura 45. Diagrama de caso de uso vendedor/ventas

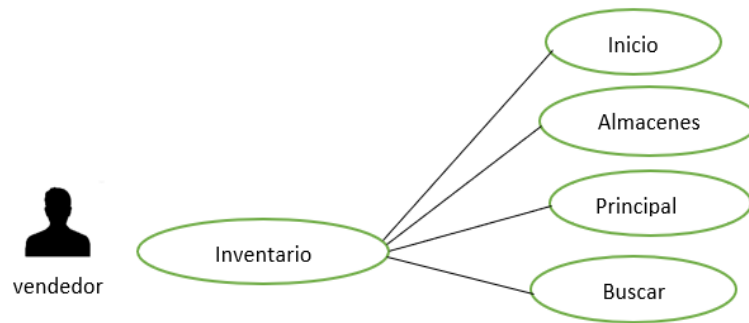


Figura 46. Diagrama de caso de uso vendedor/inventario

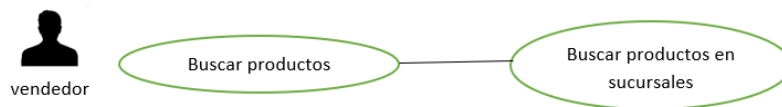


Figura 47. Diagrama de caso de uso vendedor/buscar producto

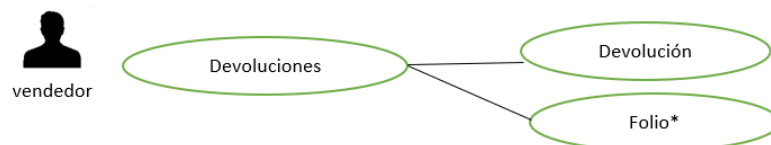


Figura 48. Diagrama de caso de uso vendedor/devoluciones

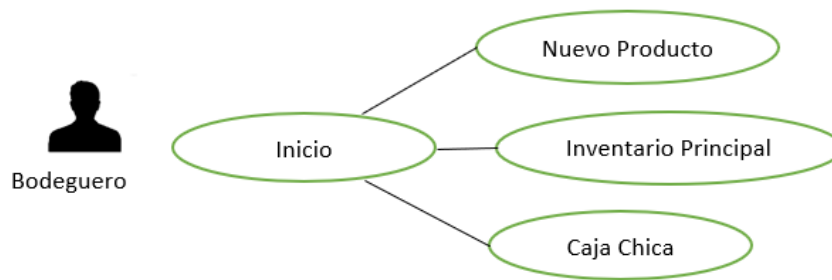


Figura 49. Diagrama de caso de uso bodeguero /inicio

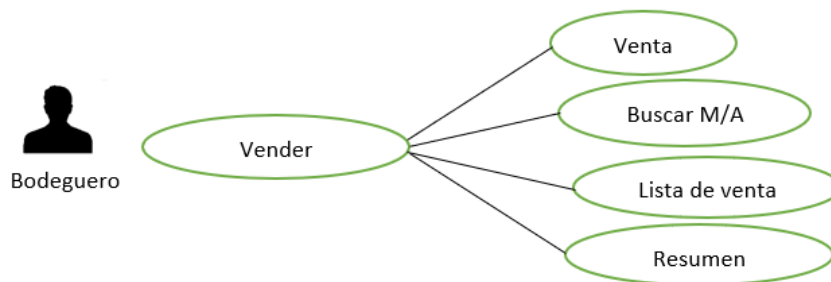


Figura 50. Diagrama de caso de uso bodeguero/vender

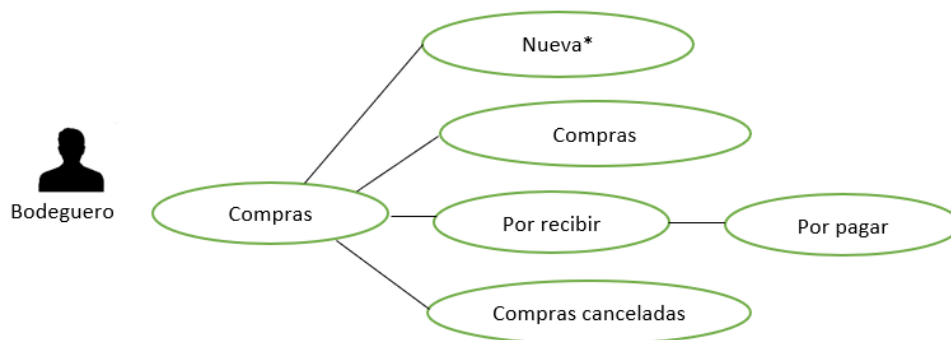


Figura 51. Diagrama de caso de uso bodeguero/ compras

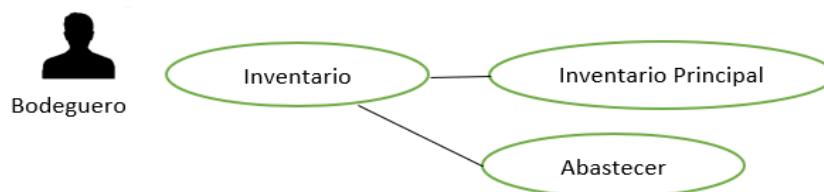
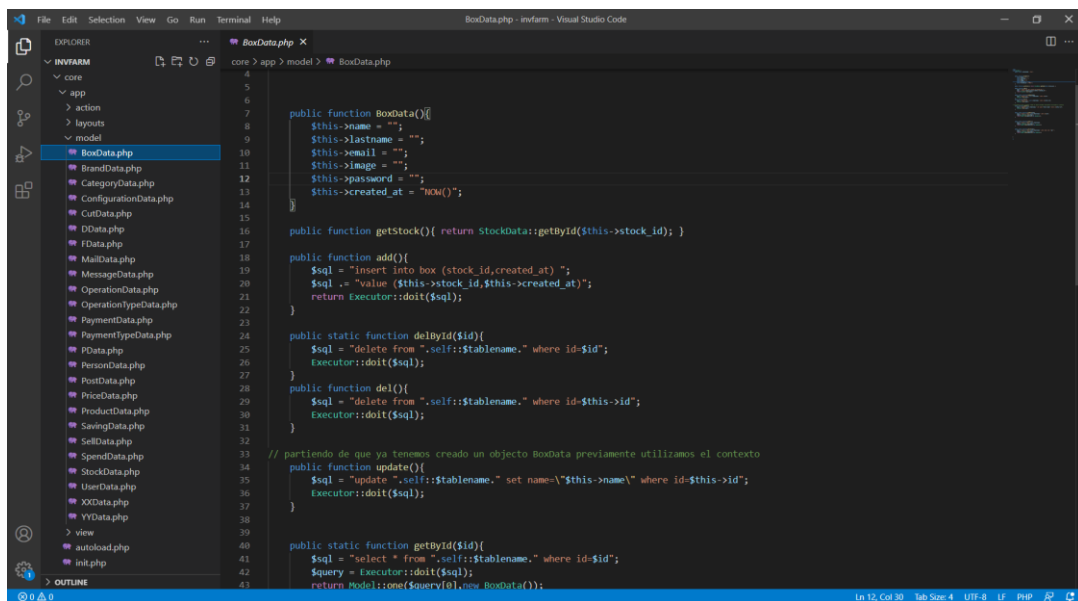


Figura 52. Diagrama de caso de uso bodeguero/inventario

Fase de codificación.

En esta fase se utilizó una arquitectura muy popular entre las aplicaciones desarrolladas utilizando el modelo de vista controlador, donde se divide la lógica de la aplicación en modelos que representen los métodos y propiedades de datos, las vistas que se encargan de procesar la petición del usuario accediendo a la información requerida, los controladores que se encargan de la entrada de eventos, al hacer uso del template que tenía las funciones del modelo en los cuales se renderiza la data que retornan las vistas. A continuación, se presentan los ejemplos de código más representativos que se desarrollaron a lo largo del proyecto.

Codificación de modelos



```
File Edit Selection View Go Run Terminal Help
BoxData.php - invfarm - Visual Studio Code
EXPLORER
  core
  app
  action
  layouts
  model
    BoxData.php
    BrandData.php
    CategoryData.php
    ConfigurationData.php
    CutData.php
    DData.php
    FData.php
    MailData.php
    MessageData.php
    OperationData.php
    OperationTypeData.php
    PaymentData.php
    PaymentTypeData.php
    PData.php
    PersonData.php
    PostData.php
    PriceData.php
    ProductData.php
    SavingData.php
    SellData.php
    SpendData.php
    StockData.php
    UserData.php
    XXData.php
    YYData.php
  view
  autoload.php
  init.php
  OUTLINE
  core > app > model > BoxData.php
  4
  5
  6
  7
  8
  9
  10
  11
  12
  13
  14
  15
  16
  17
  18
  19
  20
  21
  22
  23
  24
  25
  26
  27
  28
  29
  30
  31
  32
  33
  34
  35
  36
  37
  38
  39
  40
  41
  42
  43
  public function BoxData()
  {
    $this->name = "";
    $this->lastname = "";
    $this->xemail = "";
    $this->ximage = "";
    $this->xpassword = "";
    $this->xcreated_at = "NOW()";
  }

  public function getStock() { return StockData::getById($this->xstock_id); }

  public function add()
  {
    $sql = "insert into box (stock_id,created_at) ";
    $sql .= "value ($this->xstock_id,$this->xcreated_at)";
    return Executor::doit($sql);
  }

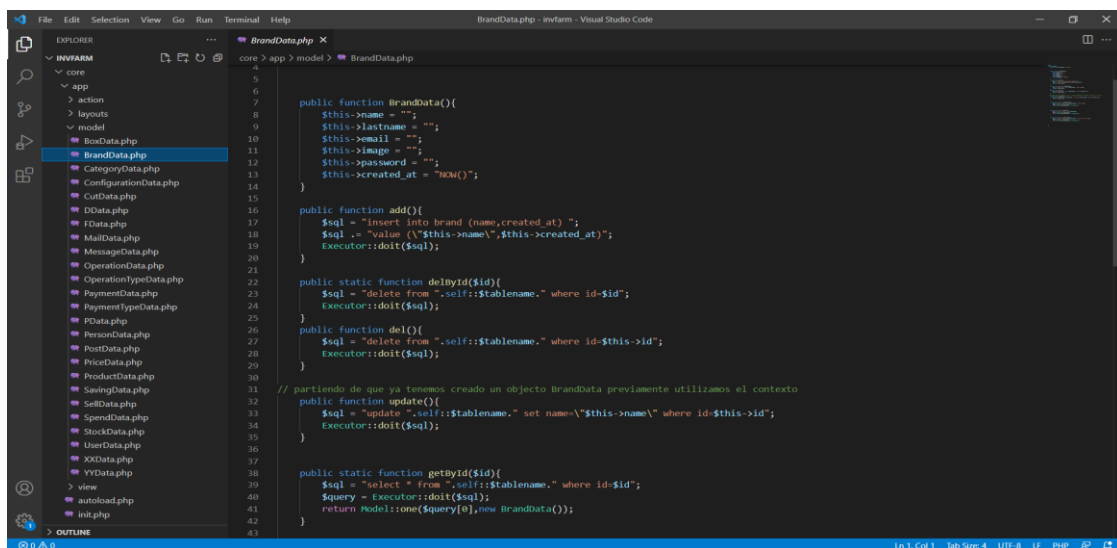
  public static function delById($id)
  {
    $sql = "delete from ".self::$tablename." where id=$id";
    Executor::doit($sql);
  }

  public function del()
  {
    $sql = "delete from ".self::$tablename." where id=$this->id";
    Executor::doit($sql);
  }

  // partiendo de que ya tenemos creado un objeto BoxData previamente utilizamos el contexto
  public function update()
  {
    $sql = "update ".self::$tablename." set name=\"$this->name\" where id=$this->id";
    Executor::doit($sql);
  }

  public static function getById($id)
  {
    $sql = "select * from ".self::$tablename." where id=$id";
    $query = Executor::doit($sql);
    return Model::one($query[0],new BoxData());
  }
  Ln 12, Col 30 Tab Size 4 UTF-8 LF PHP
```

Figura 53. Codificación de modelos



```
File Edit Selection View Go Run Terminal Help
BrandData.php - invfarm - Visual Studio Code
EXPLORER
  core
  app
  action
  layouts
  model
    BrandData.php
    CategoryData.php
    ConfigurationData.php
    CutData.php
    DData.php
    FData.php
    MailData.php
    MessageData.php
    OperationData.php
    OperationTypeData.php
    PaymentData.php
    PaymentTypeData.php
    PData.php
    PersonData.php
    PostData.php
    PriceData.php
    ProductData.php
    SavingData.php
    SellData.php
    SpendData.php
    StockData.php
    UserData.php
    XXData.php
    YYData.php
  view
  autoload.php
  init.php
  OUTLINE
  core > app > model > BrandData.php
  4
  5
  6
  7
  8
  9
  10
  11
  12
  13
  14
  15
  16
  17
  18
  19
  20
  21
  22
  23
  24
  25
  26
  27
  28
  29
  30
  31
  32
  33
  34
  35
  36
  37
  38
  39
  40
  41
  42
  43
  public function BrandData()
  {
    $this->name = "";
    $this->lastname = "";
    $this->xemail = "";
    $this->ximage = "";
    $this->xpassword = "";
    $this->xcreated_at = "NOW()";
  }

  public function add()
  {
    $sql = "insert into brand (name,created_at) ";
    $sql .= "value (\"$this->name\",$this->xcreated_at)";
    Executor::doit($sql);
  }

  public static function delById($id)
  {
    $sql = "delete from ".self::$tablename." where id=$id";
    Executor::doit($sql);
  }

  public function del()
  {
    $sql = "delete from ".self::$tablename." where id=$this->id";
    Executor::doit($sql);
  }

  // partiendo de que ya tenemos creado un objeto BrandData previamente utilizamos el contexto
  public function update()
  {
    $sql = "update ".self::$tablename." set name=\"$this->name\" where id=$this->id";
    Executor::doit($sql);
  }

  public static function getById($id)
  {
    $sql = "select * from ".self::$tablename." where id=$id";
    $query = Executor::doit($sql);
    return Model::one($query[0],new BrandData());
  }
  Ln 1, Col 1 Tab Size 4 UTF-8 LF PHP
```

Figura 54. Codificación de los modelos

```

14 }
15
16 public function add(){
17     $sql = "insert into category (name,created_at) ";
18     $sql .= "value (\\"$this->name\\",\\"$this->created_at\\");
19     Executor::doit($sql);
20 }
21
22 public static function delById($id){
23     $sql = "delete from ".self::$tablename." where id=$id";
24     Executor::doit($sql);
25 }
26 public function del(){
27     $sql = "delete from ".self::$tablename." where id=$this->id";
28     Executor::doit($sql);
29 }
30
31 // partiendo de que ya tenemos creado un objeto CategoryData previamente utilizamos el contexto
32 public function update(){
33     $sql = "update ".self::$tablename." set name=\\"$this->name\\" where id=$this->id";
34     Executor::doit($sql);
35 }
36
37 public static function getById($id){
38     $sql = "select * from ".self::$tablename." where id=$id";
39     $query = Executor::doit($sql);
40     return Model::one($query[0],new CategoryData());
41 }
42
43 public static function getAll(){
44     $sql = "select * from ".self::$tablename;
45     $query = Executor::doit($sql);
46     return Model::many($query[0],new CategoryData());
47 }
48
49 public static function getlike($o){
50 }
51
52
53

```

Figura 55. Codificación de los modelos

```

4
5 public function ConfigurationData(){
6     $this->title = "";
7     $this->content = "";
8     $this->image = "";
9     $this->user_id = "";
10    $this->is_public = "0";
11    $this->created_at = "NOW()";
12 }
13
14 public function add(){
15     $sql = "insert into ".self::$tablename." (name,short_name,is_active) ";
16     $sql .= "value (\\"$this->name\\",\\"$this->short_name\\",\\"$this->is_active\\");
17     Executor::doit($sql);
18 }
19
20 public static function delById($id){
21     $sql = "delete from ".self::$tablename." where id=$id";
22     Executor::doit($sql);
23 }
24 public function del(){
25     $sql = "delete from ".self::$tablename." where id=$this->id";
26     Executor::doit($sql);
27 }
28
29 // partiendo de que ya tenemos creado un objeto ConfigurationData previamente utilizamos el contexto
30 public function update(){
31     $sql = "update ".self::$tablename." set name=\\"$this->name\\",short_name=\\"$this->short_name\\",is_active=\\"$this->is_active\\" wher";
32     Executor::doit($sql);
33 }
34
35 public static function updateValFromName($name,$val){
36     $sql = "update ".self::$tablename." set val=\\"$val\\" where short=\\"$name\\"";
37     Executor::doit($sql);
38 }
39
40 public static function getById($id){
41     $sql = "select * from ".self::$tablename." where id=$id";
42     $query = Executor::doit($sql);
43 }
44

```

Figura 56. Codificación de modelos

```

54     if($day==3day_of_cut){
55         $breakp = $i;
56     }
57     }
58     return $breakp;
59     }
60     }
61     public static function DayCut(){
62         $breakp = self::remainDays();
63         if($breakp>1){
64             echo "<p class='alert alert-info'><i class='glyphicon glyphicon-time'></i> Faltan <b>".$breakp."</b> dias para la fecha de co";
65         }
66         }else if($breakp==1){
67             echo "<p class='alert alert-info'><i class='glyphicon glyphicon-time'></i> Se le recuerda que este es el ultimo dia que se tr";
68         }
69         }else if($breakp==0){
70             echo "<p class='alert alert-warning'><i class='glyphicon glyphicon-time'></i> Se le recuerda que se esta trabajando sobre un";
71         }
72         }
73     }
74     }
75     public static function getCut(){
76         self::DayCut();
77         if(self::getcurrent()!=null){
78             echo "<p class='alert alert-success'> Se esta trabajando sobre el corte iniciado la fecha (AAAA-MM-DD HH:MM:SS): <b>".self::g";
79         }
80         }else{
81             echo "<p class='alert alert-warning'>Bienvenido al sistema, para iniciar debes crear un corte, se ha configurado para que lo";
82         }
83     }
84     }
85     public static function getcurrent(){
86         $sql = "select * from ".self::$tablename." where finished_at is NULL";
87         $query = Executor::doit($sql);
88         $found = null;
89         $data = new CutData();
90         while($r = $query[0]->fetch_array()){
91             $data->id = $r['id'];
92             $data->finished_at = $r['finished_at'];

```

Figura 57. Codificación de modelos

Codificación de vistas

```

1 <?php
2
3 if(count($_POST)>0){
4     $user = new PersonData();
5     $user->no = $_POST["no"];
6     $user->name = $_POST["name"];
7     $user->lastname = $_POST["lastname"];
8     $user->address1 = $_POST["address1"];
9     $user->email = $_POST["email"];
10    $user->phone1 = $_POST["phone1"];
11    $user->credit_limit = $_POST["credit_limit"];
12
13
14    $user->is_active_access = isset($_POST["is_active_access"])?1:0;
15    $user->has_credit = isset($_POST["has_credit"])?1:0;
16    $user->password = sha1(md5($_POST["password"]));
17
18    $user->add_client();
19
20    print "<script>window.location='index.php?view=clients';</script>";
21
22 }
23
24
25
26

```

Figura 58. Codificación de vistas

```

1 <?php
2
3 if(count($_POST)>0){
4     $user = new PersonData();
5     $user->name = $_POST["name"];
6     $user->lastname = $_POST["lastname"];
7     $user->address1 = $_POST["address1"];
8     $user->email = $_POST["email"];
9     $user->phone1 = $_POST["phone1"];
10    $user->add_contact();
11
12    print "<script>window.location='index.php?view=contacts';</script>";
13
14 }
15
16
17
18

```

Figura 59. Codificación de vistas

```

1 <?php
2
3 if(count($_POST)>0){
4     $user = new PersonData();
5     $user->name = $_POST["name"];
6     $user->lastname = $_POST["lastname"];
7     $user->address1 = $_POST["address1"];
8     $user->email = $_POST["email"];
9     $user->phone1 = $_POST["phone1"];
10    $user->credit_limit = $_POST["credit_limit"];
11
12
13    $user->is_active_access = isset($_POST["is_active_access"])?1:0;
14    $user->has_credit = isset($_POST["has_credit"])?1:0;
15    $user->password = sha1(md5($_POST["password"]));
16
17    $user->add_client();
18
19    print "<script>window.location='index.php?view=clients';</script>";
20
21 }
22
23
24
25

```

Figura 60. Codificación de vistas

```

1 <?php
2
3 if(count($_POST)>0){
4     $product = new ProductData();
5     $product->kind = $_POST["kind"];
6     $product->code = $_POST["code"];
7     $product->barcode = $_POST["barcode"];
8     $product->name = $_POST["name"];
9     $product->price_in = $_POST["price_in"];
10    $product->price_out = $_POST["price_out"];
11    $product->unit = $_POST["unit"];
12    $product->description = $_POST["description"];
13    $product->presentation = $_POST["presentation"];
14
15    $product->width = $_POST["width"];
16    $product->height = $_POST["height"];
17    $product->weight = $_POST["weight"];
18
19    $product->expire_at = $_POST["expire_at"];
20    // $product->inventory_min = $_POST["inventory_min"];
21
22    $product->brand_id=$_POST["brand_id"]!=""?$POST["brand_id"]:"NULL";
23    $product->category_id=$_POST["category_id"]!=""?$POST["category_id"]:"NULL";
24    $product->inventory_min=$_POST["inventory_min"]!=""?$POST["inventory_min"]:"10";
25
26    // $product->category_id=$_POST["category_id"];
27    // $product->inventory_min=$_POST["inventory_min"];
28    $product->user_id = $_SESSION["user_id"];
29
30
31 if(isset($_FILES["image"])){
32     $image = new Upload($_FILES["image"]);
33     if($image->uploaded){
34         $image->process("storage/products/");
35         if($image->processed){
36             $product->image = $image->file_dst_name;
37         }
38     }
39 }
40

```

Figura 61. Codificación de vistas

```

1 <?php
2
3 if(count($_POST)>0){
4     $user = new Persondata();
5     $user->no = $_POST["no"];
6     $user->name = $_POST["name"];
7     $user->lastname = $_POST["lastname"];
8     $user->address1 = $_POST["address1"];
9     $user->email = $_POST["email"];
10    $user->phone1 = $_POST["phone1"];
11    $user->add_provider();
12
13    print "<script>>window.location='index.php?view=providers';</script>";
14
15 }
16
17
18
19

```

Figura 62. Codificación de vistas

```

1 <?php
2
3 if(isset($_POST["q"]) && !is_numeric($_POST["q"])){
4     Core::alert("Valor invalido!");
5     Core::redirect("../view-sell");
6 }
7
8 if(!isset($_SESSION["cart"])){
9
10
11
12 $product = array("product_id"=>$_POST["product_id"],"q"=>$_POST["q"]);
13 $_SESSION["cart"] = array($product);
14
15
16 $cart = $_SESSION["cart"];
17
18 ////////////////////////////////////////////////////////////////////
19 $num_succ = 0;
20 $process=false;
21 $errors = array();
22 foreach($cart as $c){
23
24     ///
25     $product = ProductData::getById($c["product_id"]);
26     $q = OperationData::getByStock($c["product_id"],stockData::getPrincipal()->id);
27     echo "q: $q";
28     //
29     if($product->xkind==2||$c["q"]<=$q){
30         $num_succ++;
31     }else{
32         $error = array("product_id"=>$c["product_id"],"message">"No hay suficiente cantidad de producto en inventario.");
33         $errors[count($errors)] = $error;
34     }
35 }
36
37 ////////////////////////////////////////////////////////////////////
38 //echo $num_succ;
39 if($num_succ==count($cart)){
40

```

Figura 63. Codificación de vistas

Codificación de controladores

```

8 class Action {
9     /**
10      * @function load
11      * @brief la funcion load carga una vista correspondiente a un modulo
12      */
13     public static function load($action){
14         // Module:$module;
15
16         if(!isset($_GET['action'])){
17             include "core/app/action/".$action.".action.php";
18         }else{
19
20             if(Action::isValid()){
21                 include "core/app/action/".$_GET['action']."-action.php";
22             }else{
23                 Action::Error("<b>404 NOT FOUND</b> Action <b>".$_GET['action']."</b> folder !! - <a href='http://evilnapsis.com/legoboo
24             }
25         }
26     }
27
28     /**
29      * @function isValid
30      * @brief valida la existencia de una vista
31      */
32     public static function isValid(){
33         $valid=false;
34         if(file_exists($file = "core/app/action/".$_GET['action']."-action.php")){
35             $valid = true;
36         }
37         return $valid;
38     }
39
40     public static function Error($message){
41         print $message;
42     }
43
44 }
45
46

```

Figura 64. Codificación de controladores

```

11  * @brief la funcion load carga un boot correspondiente a un modulo
12  */
13  public static function load($view){
14      // Module::$module;
15      if(!isset($_GET['view'])){
16          include "core/modules/".$Module::$module."/boot/".$_GET['view']."/boot-default.php";
17      }else{
18
19
20          if(self::isValid()){
21              $fullpath = "core/modules/".$Module::$module."/boot/".$_GET['view']."/boot-default.php";
22              include $fullpath;
23          }else{
24              self::Error("<b>404 NOT FOUND</b> Boot <b>".$_GET['view']."</b> folder !!");
25          }
26      }
27
28  /**
29  * @function isValid
30  * @brief valida la existencia de una vista
31  */
32  public static function isValid(){
33      $valid=false;
34      if(file_exists($file = "core/modules/".$Module::$module."/boot/".$_GET['view']."/boot-default.php")){
35          $valid = true;
36      }
37      return $valid;
38  }
39
40  public static function Error($message){
41      print $message;
42  }
43  }
44  }
45  }
46  }
47  }
48  }
49  }

```

Figura 65. Codificación de controladores

```

246  * options: '', 'ssl' or 'tls'
247  * @var string
248  */
249  public $SMTPSecure = '';
250
251  /**
252  * Whether to enable TLS encryption automatically if a server supports it,
253  * even if 'SMTPSecure' is not set to 'tls'.
254  * Be aware that in PHP 5.6 this requires that the server's certificates are valid.
255  * @var boolean
256  */
257  public $SMTPAutoTLS = true;
258
259  /**
260  * Whether to use SMTP authentication.
261  * Uses the Username and Password properties.
262  * @var boolean
263  * @see PHPMailer::$Username
264  * @see PHPMailer::$Password
265  */
266  public $SMTPAuth = false;
267
268  /**
269  * Options array passed to stream_context_create when connecting via SMTP.
270  * @var array
271  */
272  public $SMTPOptions = array();
273
274  /**
275  * SMTP username.
276  * @var string
277  */
278  public $username = '';
279
280  /**
281  * SMTP password.
282  * @var string
283  */
284  public $password = '';
285

```

Figura 66. Codificación de controladores

```

File Edit Selection View Go Run Terminal Help
Core.php - invfarm - Visual Studio Code

EXPLORER
INVIFARM
  > layouts
  > model
  > view
  > autoload.php
  > init.php
  > controller
  > forms
  > PHPExcel Classes
  > phpqrcode
  > PhpWord
  > Action.php
  > Bootload.php
  > class.phpmailer.php
  > class.pop3.php
  > class.smtp.php
  > class.upload.php
  > Cookie.php
  > Core.php
  > Database.php
  > Executor.php
  > Form.php
  > Get.php
  > Lb.php
  > Model.php
  > Module.php
  > Post.php
  > Request.php
  > RequestData.php
  > Session.php
  > View.php
  > autoload.php
  > fpdf
  > plugins
  > OUTLINE

core > controller > Core.php
11 public static $debug_sql = false;
12
13 public static $email_user = "";
14 public static $email_password = "";
15
16 public static $pdf_footer = "Generado por el Sistema de Inventario";
17 public static $email_footer = "Correo generado Automaticamente por el Sistema de Inventario";
18
19 public static $pdf_table_fillcolor = "[100, 100, 100]";
20 public static $pdf_table_column_fillcolor = "255";
21 public static $send_alert_emails = true; // enviar correos de alerta (ventas,abastecimientos, etc) -> MailData->send()
22
23 public static $discount_method = 2; // metodo de descuento 1 normal fijo, 2. porcentual
24
25 public static function includeCSS(){
26     $spath = "res/css/";
27     $handle=opendir($spath);
28     if($handle){
29         while (false !== ($entry = readdir($handle))) {
30             if($entry!="." && $entry!=".."){
31                 $fullpath = $spath.$entry;
32                 if(!is_dir($fullpath)){
33                     echo "<link rel='stylesheet' type='text/css' href='".$fullpath."' />";
34                 }
35             }
36         }
37         closedir($handle);
38     }
39 }
40
41 public static function alert($text){
42     echo "<script>alert('".$text."');</script>";
43 }
44
45 public static function redir($url){
46     echo "<script>>window.location='".$url."';</script>";
47 }
48
49
50
Ln 1, Col 1 Tab Size 4 UTF-8 LF PHP

```

Figura 67. Codificación de controladores

```

File Edit Selection View Go Run Terminal Help
Database.php - invfarm - Visual Studio Code

EXPLORER
INVIFARM
  > layouts
  > model
  > view
  > autoload.php
  > init.php
  > controller
  > forms
  > PHPExcel Classes
  > phpqrcode
  > PhpWord
  > Action.php
  > Bootload.php
  > class.phpmailer.php
  > class.pop3.php
  > class.smtp.php
  > class.upload.php
  > Cookie.php
  > Core.php
  > Database.php
  > Executor.php
  > Form.php
  > Get.php
  > Lb.php
  > Model.php
  > Module.php
  > Post.php
  > Request.php
  > RequestData.php
  > Session.php
  > View.php
  > autoload.php
  > fpdf
  > plugins
  > OUTLINE

core > controller > Database.php
1 <?php
2 class Database {
3     public static $db;
4     public static $con;
5     function Database(){
6         $this->user="root";$this->pass="";$this->host="localhost";$this->ddb="invfarmdb";
7         //$this->user="evlfnaps_admin";$this->pass="100lapalooza";$this->host="localhost";$this->ddb="evlfnaps_pvm";
8     }
9
10    function connect(){
11        $con = new mysqli($this->host,$this->user,$this->pass,$this->ddb);
12        $con->query("set sql_mode=''");
13        return $con;
14    }
15
16    public static function getCon(){
17        if(self::$con==null && self::$db==null){
18            self::$db = new Database();
19            self::$con = self::$db->connect();
20        }
21        return self::$con;
22    }
23 }
24
25
26
Ln 1, Col 1 Tab Size 4 UTF-8 LF PHP

```

Figura 68. Codificación de controladores

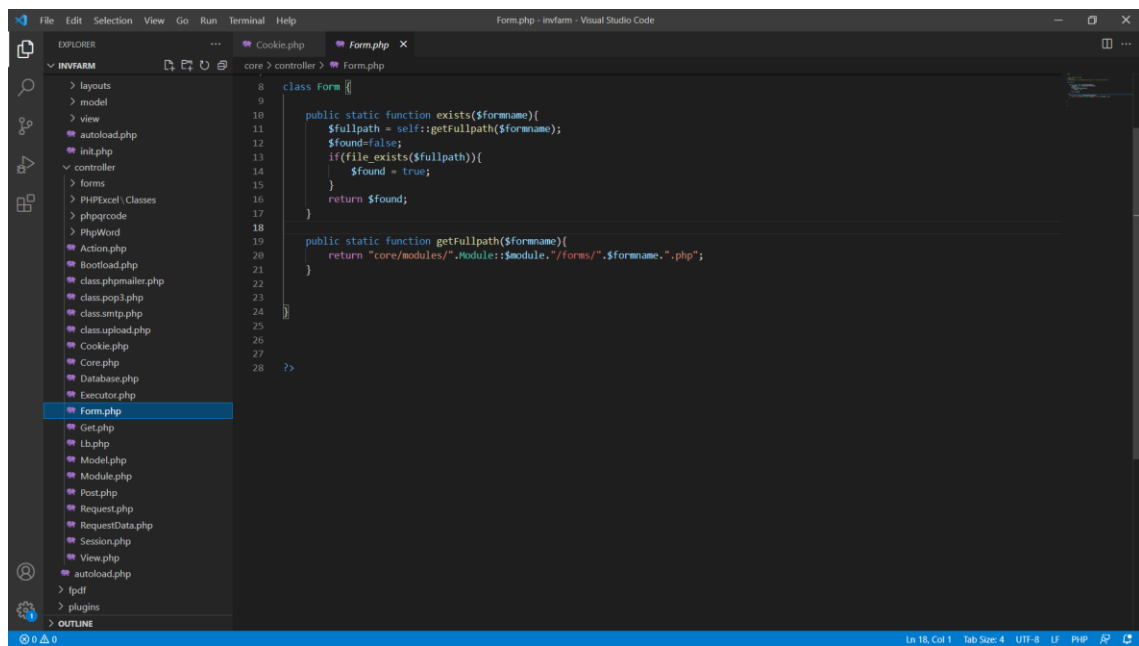


Figura 69. Codificación de controladores

Fase de pruebas.

Esta fase se adoptó en el proyecto por ser fundamental de la metodología XP para mantener un desarrollo íntegro y ágil del software, en este punto se realizó pruebas ingresando datos desde el sitio web en área local, se probó los modelos, métodos y funciones con más importancia dentro del sistema y se corroboró que el funcionamiento estaba de acorde a lo esperado. Además, se llevó a cabo pruebas de aceptación juntamente con la administradora del hospital, donde se obtuvo resultados exitosos en todos los escenarios planteados.

Historial de seguimiento de ejecución de pruebas.

Tabla 109.

Historial de seguimiento de ejecución de pruebas

Escenario	Nro.	Resultado esperado	Resultado de la prueba	
Login del sistema	1	Ingreso al sistema correcto.	Exitosa	
	2	Ingreso al sistema incorrecto.	Exitosa	
Botones de Acceso rápido	3	Ingreso a un nuevo producto.	Exitosa	
	4	Ingreso al inventario principal.	Exitosa	
		Ingreso a la caja chica.	Exitosa	
Módulo de notificaciones	5	Mensaje de notificaciones con acciones	Exitosa	
Módulo vender	6	Búsqueda de producto por nombre o código	Exitosa	
	7	Muestra lista de ventas, descarga del archivo PDF.	Exitosa	
	8	Muestra lista de ventas a crédito, descarga del archivo PDF.	Exitosa	
	9	Muestra el listado de ventas por entregar, descarga del archivo PDF.	Exitosa	
	Módulo Ventas	10	Muestra el listado de ventas por cobrar, descarga del archivo PDF.	Exitosa
		11	Muestra el listado de ventas canceladas, descarga del archivo PDF.	Exitosa
	12	Permite realizar nueva cotización.	Exitosa	
	13	Permite reabastecer el inventario	Exitosa	
	14	Muestra el listado de las compras, descarga del archivo PDF	Exitosa	
Módulo de compras	15	Muestra el listado de compras por recibir, descarga del archivo PDF.	Exitosa	
	16		Exitosa	

		Muestra el listado compras por pagar, descarga del archivo PDF	
	17	Muestra el listado de compras canceladas, descarga del archivo PDF.	Exitosa
	18	Funcionalidad del botón agregar producto.	Exitosa
	19	Muestra todos los productos de M/A, descarga del archivo PDF.	Exitosa
	20	Muestra lista de categorías.	Exitosa
	21	Funcionalidad del botón nueva categoría.	Exitosa
	22	Funcionalidad del botón productos.	Exitosa
	23	Funcionalidad de los botones editar y eliminar.	Exitosa
	24	Muestra lista de las marcas.	Exitosa
	25	Funcionalidad buscar.	Exitosa
	26	Funcionalidad del botón nueva marca.	Exitosa
Módulo catálogos	27	Funcionalidad de los botones editar y eliminar.	Exitosa
	28	Muestra el directorio de clientes.	Exitosa
	29	Funcionalidad del botón nuevo cliente.	Exitosa
	30	Funcionalidad de los botones editar y eliminar.	Exitosa
	31	Descargar del archivo PDF.	Exitosa
	32	Muestra el directorio de proveedores.	Exitosa
	33	Funcionalidad de botones, nuevo proveedor, descargar, editar y eliminar.	Exitosa
	34	Funcionalidad de los botones de la creación de sucursales, hacer principal, editar y eliminar.	Exitosa
	35	Muestra el listado para administrar precios de venta por sucursal y sus funciones.	Exitosa

	36	Muestra los contactos, y cumple con sus funciones.	Exitosa
Módulo de herramientas	37	Muestra la bandeja de entrada conjunto con los chats, cumple con sus funciones.	Exitosa
	38	Muestra la lista de los clientes con crédito, cumple con sus funciones.	Exitosa
	39	Muestra el balance básico del sistema. Muestra los gastos del sistema y	Exitosa
Módulo de finanzas	40	permite realizar las funciones especificadas.	Exitosa
	41	Muestra el listado de la caja chica, tiene funcionalidad las funciones especificadas.	Exitosa
	42	Muestra las ventas realizadas y cumple con sus funcionalidades especificadas.	Exitosa
	43	Se muestra el inventario principal conjuntamente con sus funcionalidades especificadas.	Exitosa
	44	Permite reabastecer el inventario por búsqueda de nombre o por código.	Exitosa
	45	Se muestra el valor del inventario conjuntamente con sus funcionalidades especificadas.	Exitosa
Módulo de inventario	46	Permite realizar la búsqueda por sucursales por código o por nombre. Permite visualizar el inventario global	Exitosa
	47	de todo el sistema conjuntamente con sus funcionalidades especificadas.	Exitosa
	48	Permite realizar el traspaso de una sucursal a otra.	Exitosa
	49		Exitosa

		Permite realizar la respectiva devolución con el folio generado por la venta.	Exitosa
	50		
		Muestra el listado de los traspasos realizados. Descarga del archivo PDF.	Exitosa
	51		
		Muestra el listado de las devoluciones realizadas. Descarga del archivo PDF.	
	52	Permite generar el reporte de inventario.	Exitosa
	53	Permite generar el reporte por categorías.	Exitosa
	54	Permite generar el reporte de ventas.	Exitosa
	55	Permite generar el reporte de compras.	Exitosa
	56	Permite generar el reporte de pagos a crédito.	Exitosa
Módulo de reportes	57	Permite generar el reporte de clientes populares.	Exitosa
	58	Permite generar el reporte de vendedores populares.	Exitosa
	59	Permite generar el reporte de productos populares.	Exitosa
	60	Permite generar nuevos usuarios con sus respectivos roles.	Exitosa
Módulo de administración		Permite ingresar a los usuarios generados y cumplir con sus roles.	Exitosa
	61	Permite configurar el modelo básico del sistema.	Exitosa
	62	Permite la importación de datos al sistema según el formato designado.	Exitosa

Pruebas de aceptación.

Tabla 110.

Ingreso correcto al sistema

PRUEBA DE ACEPTACIÓN	
Número: 1	N.º Historia de usuario: 1
Historia de usuario: Acceso al sistema de administración	
Nombre caso de prueba: Ingreso correcto al sistema	
Descripción: Se realiza el ingreso al sistema por medio de dos campos, el identificador del usuario y contraseña	
Condiciones de ejecución: El usuario debe tener credenciales correspondientes al rol que se le haya asignado.	
Entrada:	
<ol style="list-style-type: none">1. El usuario accede como Administrador, vendedor o bodeguero.2. El usuario debe de ingresar su nombre de usuario en el primer campo.3. El usuario debe de ingresar su contraseña en el segundo campo.4. El usuario da clic en el botón acceder.	
Resultado esperado:	
<ol style="list-style-type: none">1. Se muestra la página de inicio, esto corresponde al tipo de usuario.	
Evaluación: El ingreso al sistema es correcto	

Tabla 111.
Ingreso incorrecto al sistema

PRUEBA DE ACEPTACIÓN	
Número: 1	N.º Historia de usuario: 2
Historia de usuario: Acceso al sistema de administración	
Nombre caso de prueba: Ingreso incorrecto al sistema	
Descripción: Se realiza el ingreso al sistema por medio de dos campos, el identificador del usuario y contraseña	
Condiciones de ejecución: El usuario debe tener credenciales correspondientes al rol que se le haya asignado.	
Entrada:	
<ol style="list-style-type: none">1. El usuario accede como Administrador, vendedor o bodeguero.2. El usuario no ingresa su nombre en el primer campo.3. El usuario no ingresa la contraseña en el segundo campo.4. El usuario no llena ninguno de los campos requeridos5. El usuario da clic en el botón acceder.	
Resultado esperado:	
<ol style="list-style-type: none">2. El sistema vuelve a cargar los espacios en blanco nuevamente.3. No se registra el inicio de sesión del usuario.	
Evaluación: El ingreso incorrecto es exitoso.	

Tabla 112.

Registro del nuevo producto

PRUEBA DE ACEPTACIÓN

Número: 2

N.º Historia de usuario: 3

Historia de usuario: Añadir un nuevo producto

Nombre caso de prueba: Registro del nuevo producto

Descripción: Se procede agregar el nuevo producto en el sistema por medio de los campos designados.

Condiciones de ejecución: El usuario debe tener todos los datos solicitados por el sistema.

Entrada:

1. El usuario debe hacer clic en el botón Nuevo producto.
2. El usuario debe escoger entre los tipos de producto o servicio.
3. El usuario debe cargar una imagen del producto hacer registrado.
4. El usuario debe colocar un código interno.
5. El usuario debe colocar un código de barras.
6. El usuario debe colocar el nombre.
7. El usuario debe escoger la categoría al igual que la marca.
8. El usuario debe colocar una descripción del producto a registrar.
9. El usuario debe colocar un precio de entrada y uno de salida.
10. El usuario debe colocar las unidades que vienen en cada producto.
11. El usuario debe colocar la presentación del producto a registrar” inyectable, cápsula, tableta”.
12. Si el usuario va a registrar una caja deberá colocar, “anchura, altura, peso”, estos datos son opcionales al momento de registrar un producto.
13. El usuario debe colocar un mínimo de inventario, “esto servirá para las alertas del sistema”.
14. El usuario debe registrar el inventario inicial.
15. El usuario debe registrar la fecha de caducidad, “servirá para la alerta del sistema”
16. Finalmente, el usuario podrá agregar el nuevo producto al sistema.

Resultado esperado:

1. Se muestra el producto registrado exitosamente.

Evaluación: La función agregar es correcta.

Tabla 113.
Realización de venta

PRUEBA DE ACEPTACIÓN	
Número: 3	N.º Historia de usuario: 4
Historia de usuario: Venta en todo el sistema	
Nombre caso de prueba: Realización de venta	
Descripción: Se procede a ingresar con los tres roles de usuario que existen, administrador, vendedor, bodeguero.	
Condiciones de ejecución: debe dejar ingresar a los usuarios continuamente, deberá realizar las ventas de cada usuario individualmente.	
Entrada:	
<ol style="list-style-type: none">1. El usuario debe colocar el nombre o código del producto.2. Se muestra el listado del producto.3. El usuario debe colocar la cantidad que se va a vender.4. Se debe desplegar una lista de venta y un resumen de la misma.5. El usuario debe colocar el número de la factura.6. El usuario debe escoger el cliente si lo hay.7. El usuario debe escoger el tipo de pago.8. El usuario debe escoger el tipo de entrega.9. El usuario debe escoger la forma de pago.10. El usuario debe colocar el descuento si lo requiere.11. El usuario debe colocar el efectivo que se le entrega.12. Finalmente, se realiza la venta.	
Resultado esperado:	
<ol style="list-style-type: none">1. Se realiza la venta con éxito.	
Evaluación: La venta se realiza correctamente.	

Tabla 114.
Realización de compra

PRUEBA DE ACEPTACIÓN

Número: 4

N.º Historia de usuario: 5

Historia de usuario: Realización de compras para abastecer el inventario.

Nombre caso de prueba: Realización de compra

Descripción: Se procede a colocarse en **compras** y se realizan las pruebas de sus funciones.

Condiciones de ejecución: Debe permitir abastecer el inventario al igual debe mostrar los productos que estén agotados.

Entrada:

1. El usuario debe colocar el nombre o código del producto.
2. Se muestra el listado del producto.
3. El usuario debe colocar la cantidad que se va a comprar.
4. Se despliega la lista del reabastecimiento y un resumen de la misma.
5. El usuario debe colocar el número de la factura.
6. El usuario debe escoger el almacén que se va abastecer.
7. El usuario debe escoger el proveedor.
8. El usuario debe escoger el tipo de pago.
9. El usuario debe escoger el tipo de entrega.
10. El usuario debe escoger la forma de pago.
11. El usuario debe colocar el efectivo que se le entrega.
12. El usuario debe colocarse en el apartado de Compras para que se muestre el listado y pueda descargarlo, de igual manera con los demás, por recibir, pagar, compras canceladas.
13. Finalmente, se realiza el abastecimiento.

Resultado esperado:

1. Se obtuvieron los resultados esperados.

Evaluación: La venta se realiza correctamente.

Tabla 115.
Módulo de catálogos

PRUEBA DE ACEPTACIÓN

Número: 5

N.º Historia de usuario: 6

Historia de usuario: Realización de prueba al módulo de catálogos.

Nombre caso de prueba: Módulo de catálogos

Descripción: Se procede a colocarse en catálogos y se realizan las pruebas de sus funciones.

Condiciones de ejecución: debe realizarse las funciones del módulo según su rol.

Entrada:

1. El usuario debe agregar nuevos productos aparte del listado que se muestra.
2. El usuario debe añadir una nueva categoría, podrá visualizar la lista de las categorías.
3. El usuario debe cambiar el nombre de la categoría en el botón editar.
4. El usuario debe eliminar la categoría seleccionada.
5. El usuario debe visualizar el listado de las marcas registradas.
6. El usuario debe buscar por nombre de las categorías registradas.
7. El usuario debe visualizar el listado del directorio de clientes.
8. El usuario debe colocarse en **Nuevo cliente** para añadir un nuevo.
9. El usuario debe colocarse en editar para modificar el cliente.
10. El usuario debe colocarse en eliminar para poder quitar el usuario.
11. El usuario debe descargar el archivo.
12. El usuario debe colocarse en proveedores para poder añadir un nuevo, también tendrá que colocarse en el botón editar para modificarlo el campo al igual que podrá eliminarlo.
13. El usuario debe colocarse en sucursales para poder añadir una nueva al igual que podrá editar y eliminar esta misma.
14. El usuario debe colocarse en administrar precios para poder modificar el precio de los productos de las diferentes sucursales.

Resultado esperado:

- 1.No presenta dificultad el módulo de catálogos.

Evaluación: las funcionalidades del módulo funcionan correctamente.

Tabla 116.
Módulo de herramientas

PRUEBA DE ACEPTACIÓN	
Número: 6	N.º Historia de usuario: 1
Historia de usuario: Realizar prueba del módulo de Herramientas.	
Nombre caso de prueba: Módulo de Herramientas.	
Descripción: Se tiene que colocarse en herramientas para proceder con las pruebas de sus funciones.	
Condiciones de ejecución: Debe realizarse las funciones del módulo según su funcionalidad como ver el directorio de clientes y la bandeja de entrada de los mensajes del sistema.	
Entrada:	
<ol style="list-style-type: none">1. El usuario debe ir a Herramientas/contactos.2. El usuario debe colocar en nuevos contactos para poder añadir un nuevo al directorio de contactos.3. El usuario deberá colocarse en editar o eliminar para poder modificar el contacto creado4. El usuario deberá colocarse en Herramientas/Mensaje.5. El usuario deberá colocarse en enviar mensaje para poder chatear con los usuarios creados.6. El usuario deberá escoger el usuario.7. El usuario deberá escribir el mensaje para enviárselo al usuario designado.8. El usuario finalmente deberá presionar en enviar mensaje	
Resultado esperado:	
<ol style="list-style-type: none">1.No presenta dificultad el módulo de Herramientas.	
Evaluación: las funcionalidades del módulo funcionan correctamente.	

Tabla 117.
Módulo de finanzas

PRUEBA DE ACEPTACIÓN	
Número: 7	N.º Historia de usuario: 1
Historia de usuario: Realizar prueba del módulo de finanzas.	
Nombre caso de prueba: Módulo de finanzas.	
Descripción: Se procede a colocarse en finanzas y se realiza las pruebas de sus funciones.	
Condiciones de ejecución: debe realizar las funciones del módulo según su funcionalidad de cada sección del mismo.	
Entrada:	
<ol style="list-style-type: none">1. El usuario deberá ir a Finanzas/Crédito.2. El usuario podrá ver el listado de los clientes con crédito.3. El usuario para poder descargar el archivo deberá colocarse en el botón generar reporte4. El usuario deberá colocarse en el botón realizar pago para cancelar el total del crédito de dicho cliente.5. El usuario para poder ver el historial de ese cliente deberá colocarse en el botón historial.6. El usuario deberá colocarse en Finanzas/Balance.7. El usuario para poder generar el balance deberá escoger una fecha y dar clic en procesar.8. El usuario deberá colocarse en Finanzas/Gastos.9. El usuario deberá añadir los gastos que sean necesarios.10. El usuario deberá colocarse en Finanzas/caja.11. El usuario deberá dar clic en procesar ventas para que se hagan efectivas en el sistema.12. El usuario deberá dar clic en historial para poder ver las ventas que fueron realizadas y pueda descargar el archivo PDF.	
Resultado esperado:	
<ol style="list-style-type: none">1. No presenta dificultad el módulo de Herramientas.2. No presenta dificultad en la descarga de los archivos.	
Evaluación: las funcionalidades del módulo funcionan correctamente en toda su totalidad.	

Tabla 118.
Módulo de inventario.

PRUEBA DE ACEPTACIÓN	
Número: 8	N.º Historia de usuario: 1
Historia de usuario: Realizar prueba del módulo de inventario.	
Nombre caso de prueba: Módulo de inventario.	
Descripción: Se procede a colocarse en inventario y se realizan las pruebas de sus funciones.	
Condiciones de ejecución: Debe realizar las funciones correspondientes de cada sección del mismo.	
Entrada:	
<ol style="list-style-type: none">1. El usuario tendrá que observar el inventario principal al igual que visualizar los productos que están con menos existencias en el inventario.2. El usuario deberá dar clic en agregar para añadir una nueva cantidad al inventario.3. El usuario deberá dar clic en quitar si desea eliminar ese producto del sistema.4. El usuario deberá dar clic en el botón historial para ver sus detalles.5. El usuario deberá dar clic en inicio para que le dirija el panel principal.6. El usuario deberá dar clic en almacenes para poder ver el inventario de cada sucursal.7. El usuario deberá dar clic en descargar para generar el archivo PDF.8. El usuario debe colocarse en reabastecer para poder agregar una nueva cantidad al inventario dándole clic en agregar.9. El usuario para poder ver el valor de los inventarios deberá dar clic en valor del inventario donde se podrá visualizar todos los inventarios del sistema.10. El usuario para poder buscar un x producto en los distintos inventarios deberá dar clic en buscar producto en sucursales.11. El usuario para poder ver el inventario global deberá dar clic en inventario global, donde podrá descargar el archivo PDF y visualizar las sucursales.12. El usuario para poder enviar un producto a otro inventario tendrá que dar clic en traspasar, donde deberá escoger la sucursal, el producto y finalmente traspasar el producto seleccionado, de igual manera tendrá que hacer con las devoluciones.13. El usuario debe dar clic en traspasos para ver el historial de estos mismo al igual que deberá realizarlo con las devoluciones.	
Resultado esperado:	
<ol style="list-style-type: none">1. No presenta dificultad el módulo de inventario.	
Evaluación: las funcionalidades del módulo funcionan correctamente en toda su totalidad.	

Tabla 119.
Módulo de reportes.

PRUEBA DE ACEPTACIÓN	
Número: 9	N.º Historia de usuario: 1
Historia de usuario: Revisión del módulo de reportes.	
Nombre caso de prueba: Módulo de Reportes.	
Descripción: Se procede a colocarse en reportes y se realiza las pruebas de sus funciones.	
Condiciones de ejecución: Debe realizarse las funciones del módulo de cada sección del mismo.	
Entrada:	
<ol style="list-style-type: none">1. El usuario deberá colocarse en Log de Inventario, donde deberá escoger la sucursal y productos al igual que la fecha para generar el reporte.2. El usuario deberá realizar la misma secuencia de pasos para generar el reporte de categorías, ventas, compras, Reporte de pagos “crédito”, clientes populares, vendedores populares, productos populares.	
Resultado esperado:	
<ol style="list-style-type: none">1. No presenta dificultad el módulo de reportes2. Se muestran todas las alarmas correspondientes como no dejar espacios en blanco para generar el reporte.	
Evaluación: las funcionalidades del módulo funcionan correctamente en toda su totalidad.	

Tabla 120.
Módulo de administración.

PRUEBA DE ACEPTACIÓN	
Número: 10	N.º Historia de usuario: 1
Historia de usuario: Realizar prueba del módulo de administración.	
Nombre caso de prueba: Módulo de administración.	
Descripción: Se procede a colocarse en administración y se realiza las pruebas de sus funciones.	
Condiciones de ejecución: Debe realizarse las funciones del módulo según las secciones del mismo.	
Entrada:	
<ol style="list-style-type: none">1. El usuario deberá hacer clic en nuevo administrador para añadir uno nuevo al listado existente, de igual manera deberá hacerlo con los dos restantes que son bodeguero y vendedor.2. El usuario deberá rellenar los datos solicitados por el sistema.3. El usuario deberá hacer clic en agregar usuario para poder visualizarlo en la lista principal4. El sistema le mostrara un mensaje “el usuario a sido añadido correctamente”5. El usuario podrá ver la lista de usuarios creados y asignados su rol.6. El usuario deberá hacer clic en editar en el nombre de usuario para poder modificar los campos de ese usuario.7. El usuario deberá hacer clic en eliminar para deshacerse de algún usuario que no esté en uso.8. El usuario tendrá que hacer clic en configuración para poder detallar algunos datos del sistema como el título del ticket, IVA, colocar el email de la institución.	
Resultado esperado:	
<ol style="list-style-type: none">1.No presenta dificultad el módulo de administración.2.Se muestran los mensajes de añadir nuevo usuario al igual que editar y eliminar.	
Evaluación: las funcionalidades del módulo funcionan correctamente en toda su totalidad.	

Anexo 13. Manual de usuario del sistema

**Manual de usuario del sistema de control de inventario para la farmacia del Hospital
del Dia Nova Salud de la ciudad de Tulcán**

Autor:

Darwin Fernando Tipan viveros

Introducción

El sistema de control de inventario para los materiales y accesorios de farmacia es una plataforma web que integra los módulos de control, ventas, compras, reportes etc., este software se enfoca en el control de las existencias de farmacia, además ofrece funcionalidades extras que son módulos básicos.

Requisitos de software

Se han establecido requisitos mínimos de software para el correcto funcionamiento, como:

1. Navegadores web: Microsoft Edge (recomendable).
2. Sistema operativo: Windows (recomendable).
3. Servidor XAMPP v7.2.11.exe. (Indispensable, diríjase a la dirección brindada para la descargar de la version especificada).

<https://drive.google.com/file/d/1ZGIvjv6N08nJ-P0Vf5TYOke6k7nzWVLZ/view?usp=sharing>

Requisitos de hardware

Se han establecido requisitos mínimos de hardware para el correcto funcionamiento, como:

1. Procesador Core i3 o superior.
2. Memoria RAM de 4gb.
3. Resolución de pantalla mínimo 1300 x 930 píxeles o superior.

Rol de Administrador.

A continuación, se hará una explicación general sobre la funcionalidad del sistema, para acceder al sistema se utilizará el siguiente enlace: <http://localhost/NOVA/>. Al ingresar a esta dirección se visualizará el formulario de inicio de sesión, en donde deberá colocar usuario y contraseña registrado.

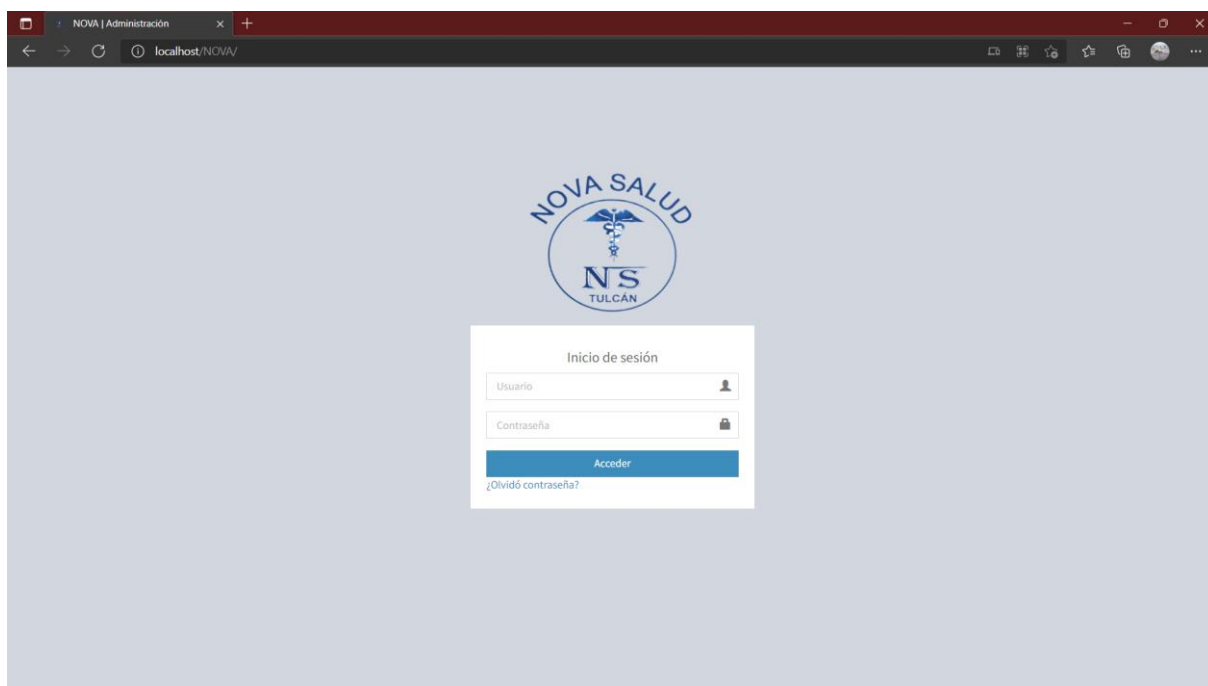


Figura 70. Inicio de sesión

Aquí tenemos la pantalla principal del sistema, en donde muestra la barra de simulación de estadística del sistema conjuntamente con sus respectivas funciones.

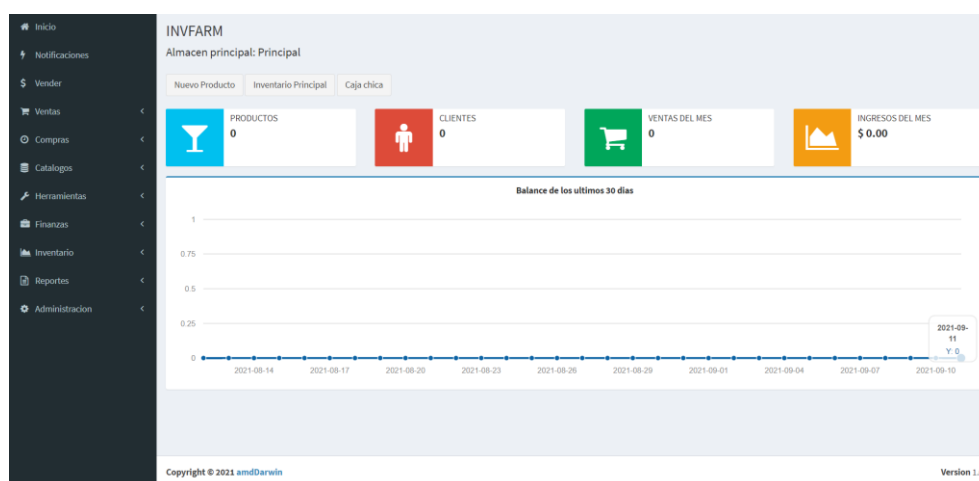


Figura 71. Panel principal del sistema

Una vez ingresado tendremos que añadir nuestras marcas y categorías en el sistema, también podremos añadir sucursales y administrar sus precios, y lo más importante, se podrá añadir a los proveedores teniendo así un mayor orden dentro del sistema.



Figura 72. Módulo de categorías

Ahora nos dirigiremos al apartado de **Nuevo Producto**, se nos mostrará la siguiente ventana, donde deberemos llenar los datos que nos solicita para agregar un nuevo producto.

Figura 73. Sección para agregar nuevo producto en el sistema

Una vez agregado el nuevo producto nos dirigiremos a **Catálogos/Productos**, en este apartado se mostrará todos los productos registrados hasta el momento.

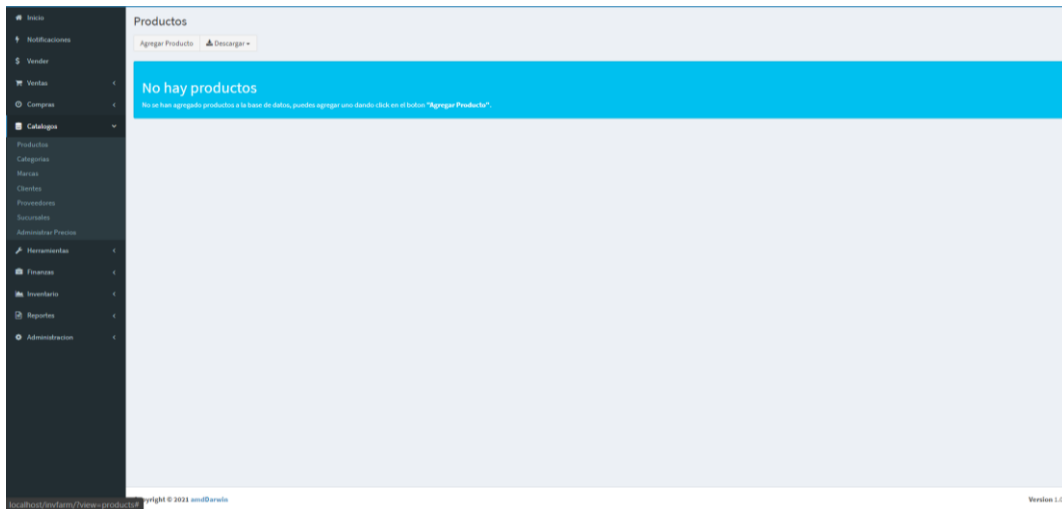


Figura 74. Sección para visualizar los productos registrados

Una vez agregado los productos nos iremos al apartado de **Inventario/Inventario Principal**, aquí se mostrarán los productos con sus existencias actuales donde podrá agregar más cantidades sin necesidad de regresar a registrar nuevo producto.

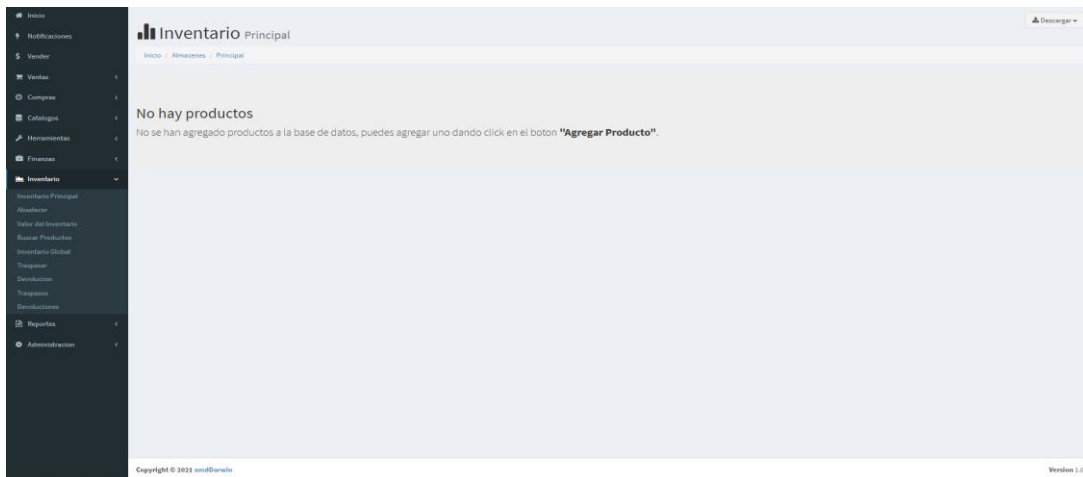


Figura 75. Módulo del inventario

Una vez registrado los productos podemos empezar a venderlos, para eso nos dirigimos al apartado de **venta**, donde podemos buscar por código o por nombre con abreviación de los productos.

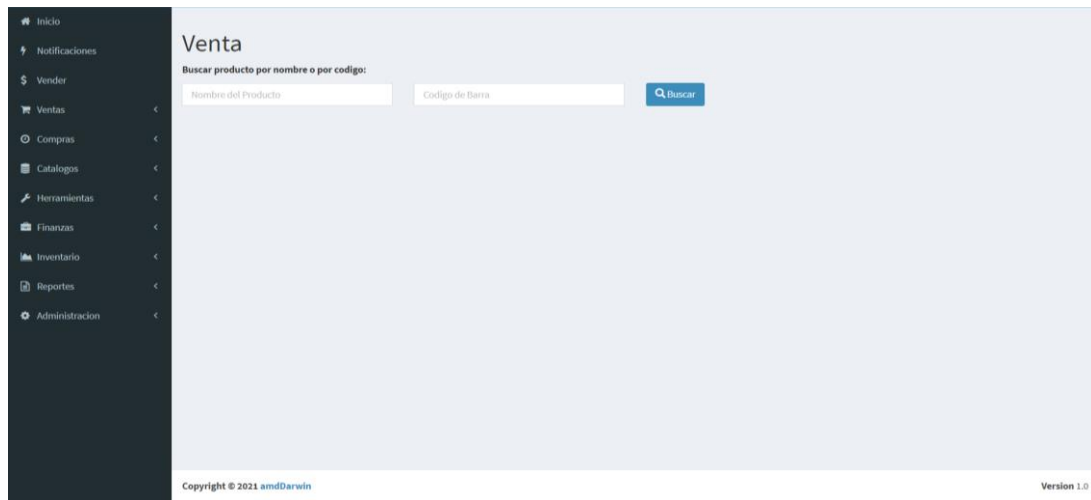


Figura 76. Sección de realización de ventas

Al realizar la venta nos tendremos que redirigir al apartado de **Finanzas**, donde se mostrará todas las ventas que se generaron en el transcurso del día, donde se nos pedirá **procesar venta** para que se haga efectiva en el sistema, este método es aplicado por petición del usuario.

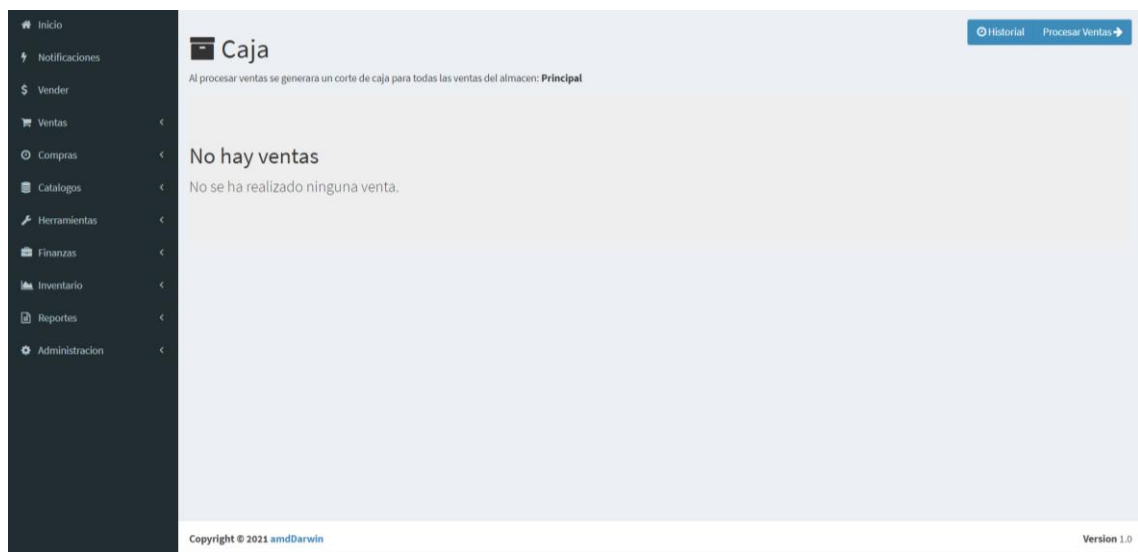


Figura 77. Sección de caja

Ahora tendremos que abastecer el inventario, nos dirigimos al apartado de **Compras/Nueva**, se nos mostrará la siguiente ventana con el mensaje de abastecer donde podremos buscar por código o por nombre abreviado.

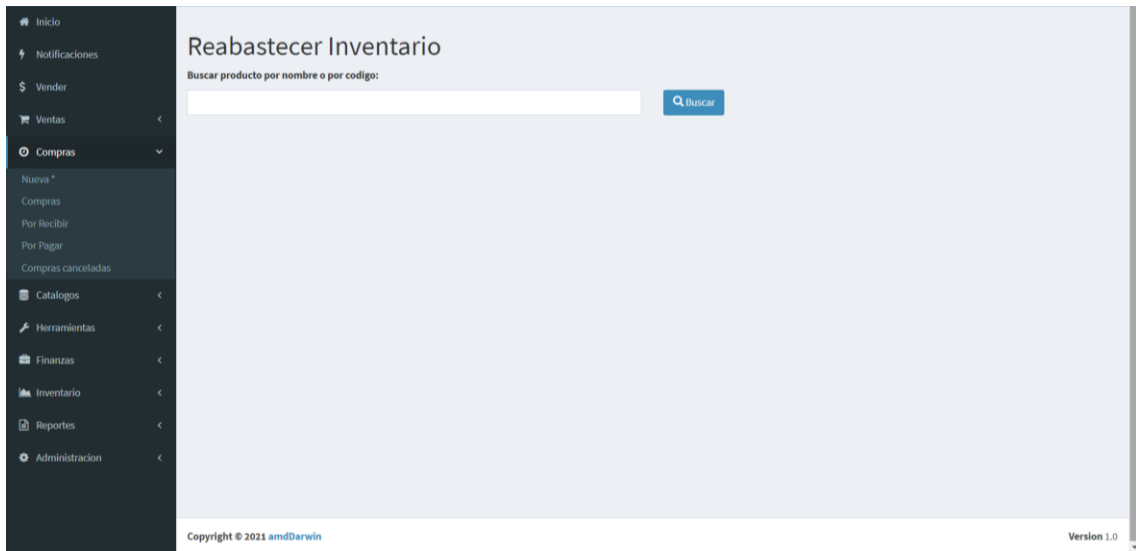


Figura 78. Sección de reabastecimiento del inventario

Al abastecer el inventario tendremos que verificar que en verdad se hizo la compra nos dirigiremos al apartado de **Compras** donde se nos mostrará el registro de todas las compras que hemos hecho actualmente en el mes o en el transcurso de la semana.

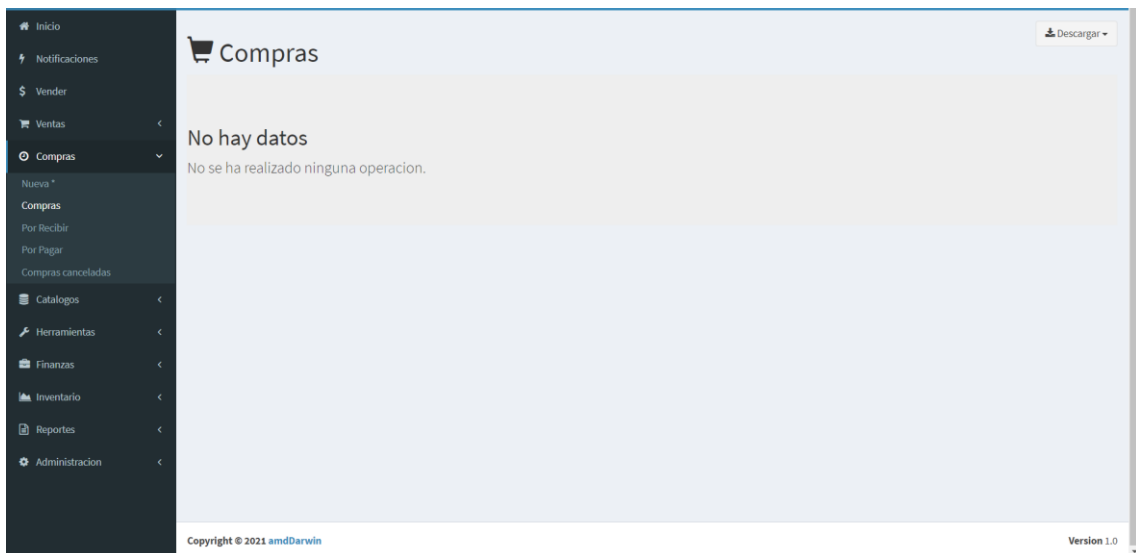


Figura 79. Sección de compras

Ahora nos dirigiremos al apartado de administración donde podremos crear los tres tipos de usuario para el manejo del sistema, donde serán designados a una sucursal específica esto con la finalidad de hacer un nuevo proceso en cuanto respecta al control de usuarios dentro de la clínica. Una vez creado los usuarios estos tendrán su propio panel de control donde estará limitado con sus funciones que solo le respecta a ese usuario específico.

Nombre completo	Nombre de usuario	Email	Almacen	Activo	Tipo	
Administrador		admdarwin		✓	Administrador	Editar
almacenista1 NN	almacenista1	admin@intelligentSociety.com	Principal	✓	Almacenista	Editar
vendedor1 nn	vendedor1	admin@intelligentSociety.com	Principal	✓	Vendedor	Editar

Figura 80. Lista de usuarios

Rol del Bodeguero.

Iniciamos sesión con las credenciales proporcionadas por el administrador, en caso que se fueran olvidadas se les enviará en un mensaje de chat a la bandeja de entrada de mensajes del sistema del **Bodeguero**.

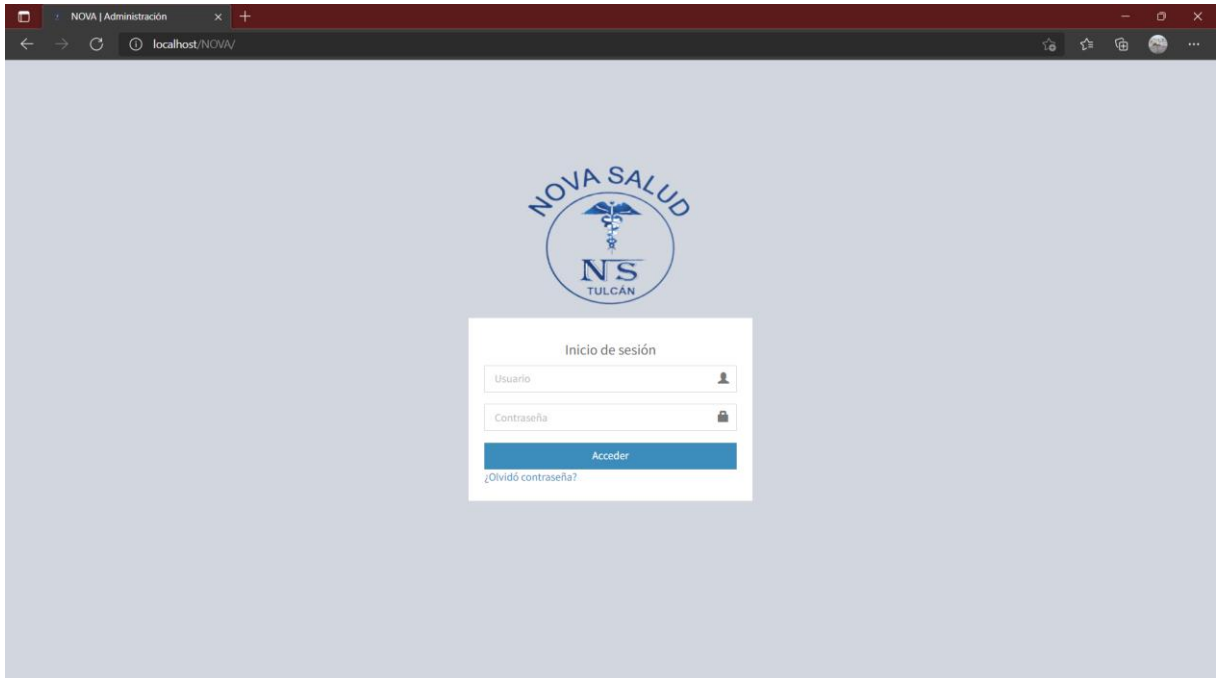


Figura 81. Inicio de sesión

Ahora se nos mostrará la ventana del bodeguero donde se podrá manipular todo el módulo correspondiente a este usuario. En esta ventana principal se nos mostrara los tres botones de acceso rápido a las funciones que cuentan con sus respectivos formularios.

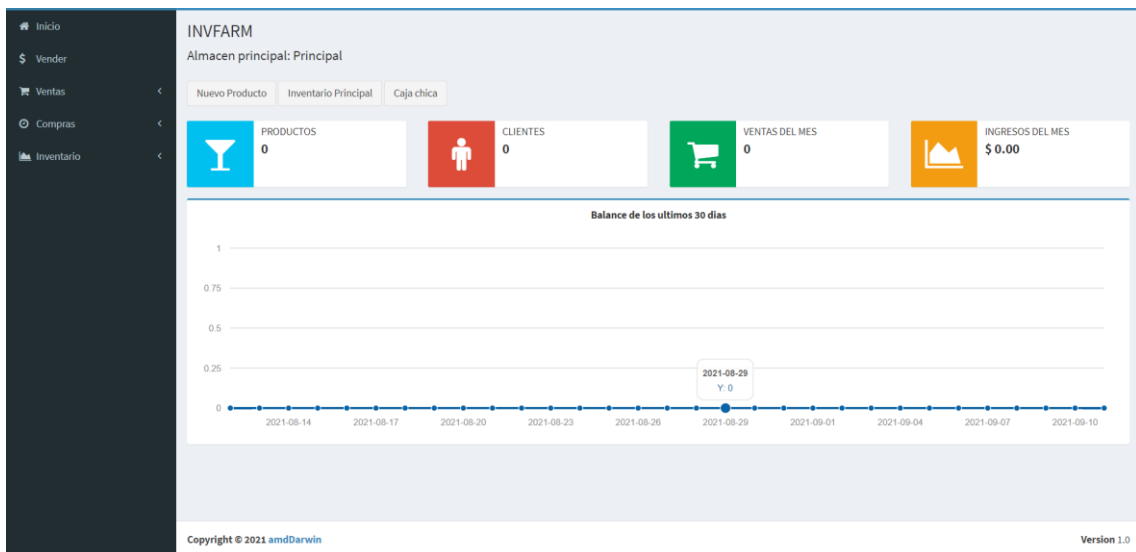


Figura 82. Panel de control

Ahora bien, dirijámonos al botón de acceso rápido, **Nuevo producto**, se nos mostrará la ventana de registro del producto, al llenar los datos solicitados por el sistema saldrá automáticamente las marcas y las categorías ya añadidas por el administrador.

The screenshot shows a web interface for adding a new product. On the left is a dark sidebar with navigation options: Inicio, Vender, Ventas, Compras, and Inventario. The main content area is titled 'Nuevo Producto' and contains a form with the following fields: 'Tipo' (dropdown set to 'Producto'), 'Imagen' (file upload button 'Elegir archivo'), 'Código Interno' (text input), 'Código de Barras' (text input), 'Nombre' (text input), 'Categoría' (dropdown set to '--NINGUNA--'), 'Marca' (dropdown set to '--NINGUNA--'), 'Descripción' (text area), 'Precio de Entrada (\$)' (text input), 'Precio de Salida (\$)' (text input), 'Unidad' (text input), 'Presentación' (text input), 'Ancho*' (text input), 'Altura*' (text input), 'Peso*' (text input), 'Mínima en Inventario' (text input), 'Inventario Inicial' (text input), and 'Fecha de caducidad' (date picker). A blue 'Agregar Producto' button is located at the bottom of the form.

Figura 83. Ventana de agregación de un nuevo producto

Una vez registrado el producto podemos verificarlo presionando en el botón inventario, donde se mostrará el nuevo producto conjuntamente con su cantidad registrada.

The screenshot shows the 'Inventario Principal' page. The sidebar on the left has 'Inventario' selected, with sub-items 'Inventario Principal' and 'Abastecer'. The main content area has a header 'Inventario Principal' with a 'Descargar' button. Below the header is a breadcrumb 'Inicio / Almacenes / Principal'. The main content area displays a message: 'No hay productos' and 'No se han agregado productos a la base de datos, puedes agregar uno dando click en el boton "Agregar Producto".' The footer contains 'Copyright © 2021 amdDarwin' and 'Version 1.0'.

Figura 84. Venta del inventario

Este usuario también tendrá habilitada la parte de ventas por si se presenta una repentina. También podrá verificar las ventas realizadas en ese momento en su panel de control, también podrá ver un sub menú donde aparecerán las ventas por entregar, cobrar, canceladas, cotizaciones, dándole así una mayor fluidez a su rol como bodeguero.

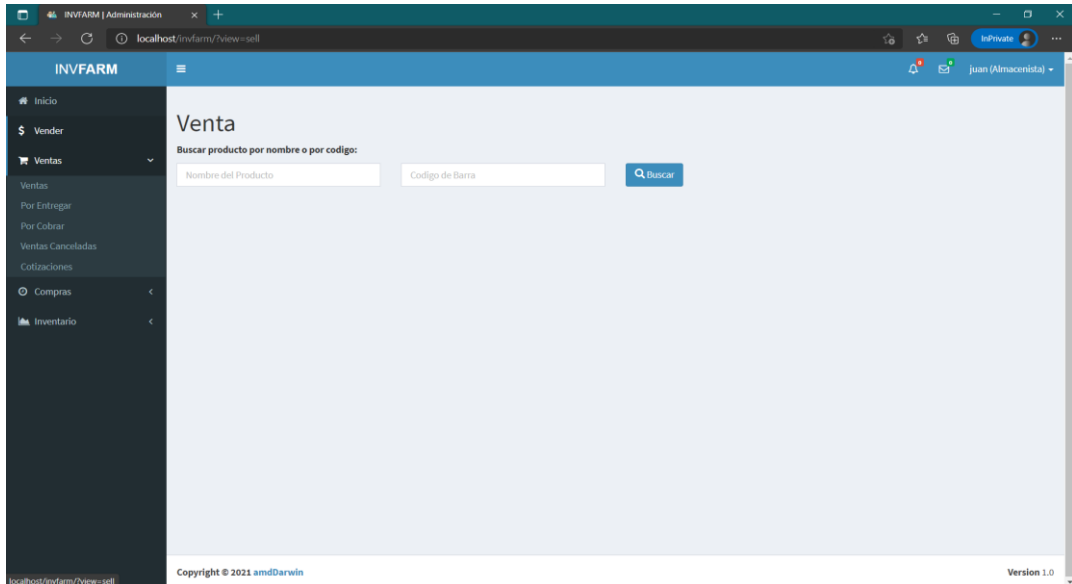


Figura 85. Ventana de ventas

Ahora tendremos que abastecer el inventario, nos dirigimos a **Compras**, se nos mostrará la siguiente ventana donde podrá agregar más cantidades a los materiales y accesorios que se encuentren con poca existencia en el inventario, una vez abastecido las podrá verificar en **Inventario** los productos que fueron reabastecidos.

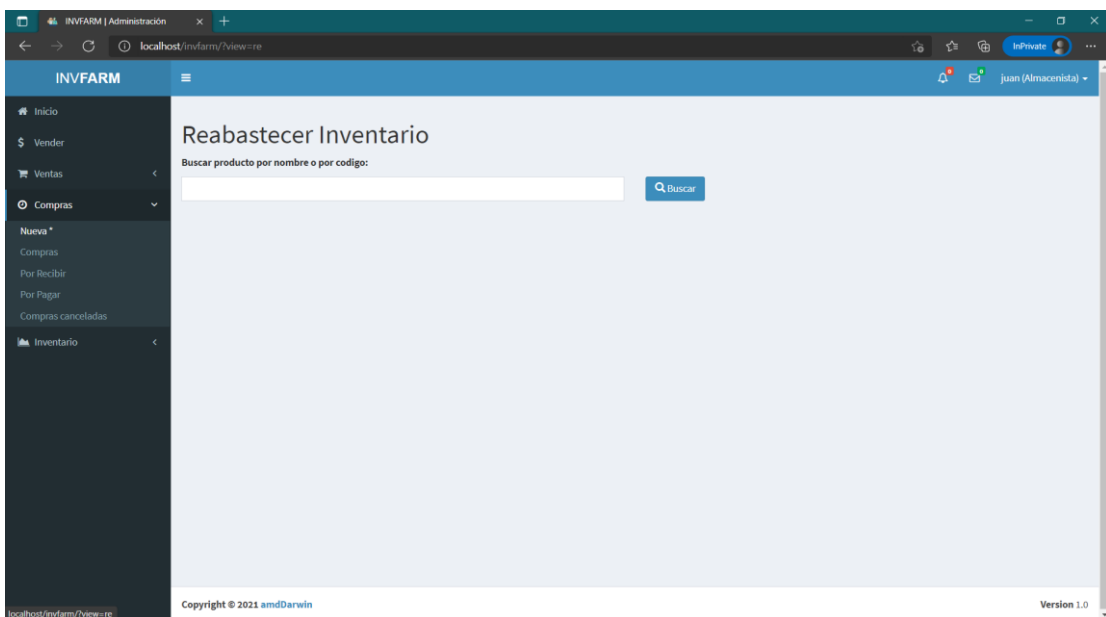


Figura 86. Ventana de reabastecimiento del inventario

Rol del Vendedor.

Iniciamos sesión con las credenciales proporcionadas por el administrador, en caso que se fueran olvidadas se las enviará en un mensaje de chat a la bandeja de entrada de mensajes del sistema del vendedor.



Figura 87. Inicio de sesión

Ahora se nos mostrará la ventana de vendedor donde se podrá manipular todo el módulo de ventas.

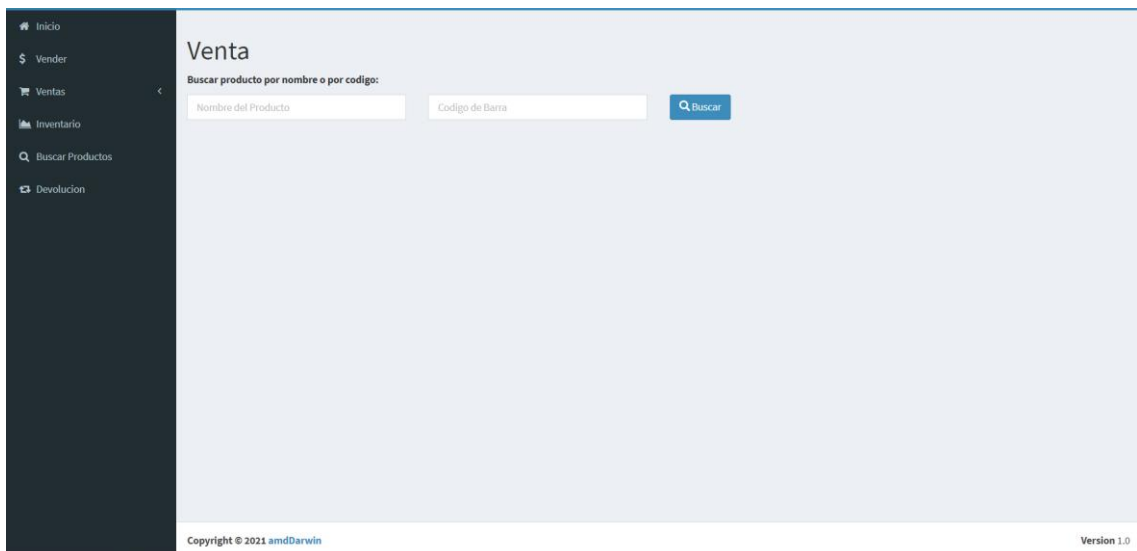


Figura 88. Ventana de ventas

El usuario podrá ver el inventario principal y al igual de las sucursales creadas hasta ese momento.



Figura 89. Ventana de inventario

También se podrá buscar productos entre las diferentes sucursales que estén en ese momento creadas.



Figura 90. Ventana de búsqueda entre sucursales

Se podrá manipular las devoluciones de las ventas que haya realizado en su panel de control, donde tendrá que colocar el folio de la venta que haya generado el sistema automáticamente.

Devolucion

Escriba el FOLIO de la venta:

Folio*

Folio

Proceder

Copyright © 2021 amdDarwin

Version 1.0

Figura 91. Ventana de devoluciones